GENOMES to LIFE
BIOLOGICAL SOLUTIONS
FOR ENERGY CHALLENGES
U.S. DEPARTMENT OF ENERGY
INNOVATIVE APPROACHES ALONG UNCONVENTIONAL PATHS

Technical Reference

Compilation and Review of Data Standards and Application to the Genomes to Life Program

January 31, 2004, v. 1.2

Contributing Members of the
GTL Data Standards Working Group

Adam Arkin[2,3,7], John Ambrosiano[4], Gyorgy Babnigg[1],
Ed Frank[1], Al Geist[5], Carol Giometti[1], Janet Jacobsen[7],
Nagiza Samatova[5], Nancy Slater[3] and Ron Taylor[6]

[1]Argonne National Laboratory, [2]Howard Hughes Medical Institute,
[3]Lawrence Berkeley National Laboratory,
[4]Los Alamos National Laboratory, [5]Oak Ridge National Laboratory,
[6]Pacific Northwest National Laboratory, [7]University of California, Berkeley

Office of Science
U.S. DEPARTMENT OF ENERGY

Table of Contents

Members of the GTL Data Standards Working Group

Eric Alm
Physical Biosciences Division
Lawrence Berkeley National Laboratory

John Ambrosiano
Computer and Computational Sciences
Los Alamos National Laboratory

Adam Arkin, *Chairperson*
Physical Biosciences Division
Lawrence Berkeley National Laboratory
Department of Bioengineering
University of California, Berkeley
Howard Hughes Medical Institute

Gyorgy Babnigg
Biosciences Division
Argonne National Laboratory

Ed Frank
Mathematics and Computer Science Division
Argonne National Laboratory

Al Geist
Computer Science and Mathematics Division
Oak Ridge National Laboratory

Carol Giometti
Biosciences Division
Argonne National Laboratory

Janet Jacobsen
Institute for Quantitative Biomedical Research
University of California, Berkeley

Eugene Kolker
BIATECH

Bertram Ludaescher
Data and Knowledge Systems
San Diego Supercomputer Center
Department of Computer Science
and Engineering
University of California, San Diego

George Michaels
Biological Sciences
Pacific Northwest National Laboratory

D. William Nguyen
Genetics Department
Harvard Medical School
Harvard University

Bahram Parvin
Computational Research Division
Lawrence Berkeley National Laboratory

Nagiza Samatova
Computer Science and Mathematics Division
Oak Ridge National Laboratory

Nancy Slater
Physical Biosciences Division
Lawrence Berkeley National Laboratory

Ron Taylor
Biological Sciences Division
Pacific Northwest National Laboratory

Ed Uberbacher
Life Sciences Division
Oak Ridge National Laboratory

Data Standards for the Genomes to Life Program

Adam Arkin[2,3,7], Carol Giometti[1], Janet Jacobsen[7],
John Ambrosiano[4], Gyorgy Babnigg[1], Ed Frank[1], Al Geist[5],
Nagiza Samatova[5], Nancy Slater[3], and Ron Taylor[6]

*[1]Argonne National Laboratory, [2]Howard Hughes Medical Institute,
[3]Lawrence Berkeley National Laboratory, [4]Los Alamos National Laboratory,
[5]Oak Ridge National Laboratory, [6]Pacific Northwest National Laboratory,
[7]University of California, Berkeley*

Contents

Introduction

Existing GTL Projects already have produced volumes of data and, over the course of the next five years, will produce an estimated hundreds, or possibly thousands, of terabytes of data from hundreds of experiments conducted at dozens of laboratories in National Labs and universities across the nation. These data will be the basis for publications by individual researchers, research groups, and multi-institutional collaborations, and the basis for future DOE decisions on funding further research in bioremediation. The short-term and long-term value of the data to project participants, to the DOE, and to the nation depends, however, on being able to access the data and on how, or whether, the data are archived.

The ability to **access** data is the starting point for data analysis and interpretation, data integration, data mining, and development of data-driven models. Limited or inefficient data access means that less data are analyzed in a cost-effective and timely manner. Data production in the GTL Program will likely outstrip, or may have already outstripped, the ability to analyze the data. Being able to access data depends on two key factors: **data standards** and **implementation of the data standards**. For the purpose of this proposal, a data standard is defined as a standard, documented way in which data and information about the data are describe. The attributes of the

Date of Last Revision: January 31, 2004

experiment in which the data were collected need to be known and the measurements corresponding to the data collected need to be described. In general terms, a data standard could be a form (electronic or paper) that is completed by a researcher or a document that prescribes how a protocol or experiment should be described in writing.

Data standards are critical to data access because they provide a framework for **organizing and managing data**. Researchers spend significant amounts of time managing data and information about experiments using lab notebooks, computer files, Excel spreadsheets, *etc*. In addition, data output format varies for different equipment and usually need to be formatted differently for the variety of computer programs used to display and analyze the data. If, however, data for a given type of experiment were converted from vendor format to a **format** defined by a data standard, then researchers and software developers could save time. In addition, if data and information describing how they were obtained were available in a consistent format throughout the GTL Program, comparison and integration of results would be facilitated and a data repository could be built to encourage project-wide data mining.

Data standards also are essential for **archiving** data sets. If data are stored together with the experiment metadata (*i.e.,* information about the data) in an 'information/data package', then the data retain their value due to the accessibility of information about measurement and analysis procedures.

DOE's commitment to developing data standards for the GTL Program is needed to ensure that the most value is obtained from DOE's expenditures on experimental work and to provide a data repository that can be used as the basis for on-going model development. By developing data standards for experiments conducted as part of the GTL Program, DOE has the opportunity to facilitate data sharing not only within the DOE community, but also with research institutes through the world.

Scope of Data Standards for the Genomes to Life Program

Given current research programs of the GTL Program, data standards need to be developed and implemented for: sample preparation and generation; and experimental methods such as flow cytometry, mass spectrometry, microarray, and 2D gels; and data processing methods. Existing, applicable data standards may be selected or adapted for use, but as described below, will need to be supplemented by a top-level experiment description and will require changes in order to eliminate inconsistencies in representation of information common to all data standards and in how data from experiments are handled.

Specifications for Data Standards

One of the most well known data standards developed in recent years is the Minimum Information About Microarray Experiments (MIAME) standard [1]. The MIAME standard focuses on information about microarray experiments, *i.e.*, array description, experiment design, sample preparation, and data processing. The MIAME standard provides an example of a collaborative approach to developing a data standard, and we support the use or adaptation of the MIAME standard in the GTL Program.

Data standards developed for the GTL Program should cover:

- **metadata**, *i.e.*, information about how the experiment or set of experiments was conducted, sample preparation and generation, **biological context** (genus, species, strain, life cycle stage, cellular location, *etc.*), and **environmental context** (site location, aerobic or anerobic conditions, pH, *etc.*);
- **raw data**, **processed data**, and/or **interpreted data,** together with how the data were processed or analyzed.

Metadata and data are stored electronically as both character and numeric data. We propose two approaches to numeric data. In one approach, numeric data is included in the same electronic file as the metadata, either in clear text or using a 'binary encoding' [2]. Alternatively, external file references are supported, so that large data files can be stored or archived as electronic files, but be referenced by the metadata file. The metadata file will not only include the name of the external file, but also will describe the data in the file, how the data were encoded, and whether or how the file was compressed. Describing the data themselves requires specifying (at least) the following information: data 'coordinates', that is, the independent variables (*e.g.*, time, dosage, pH, replicate number, *etc.*); the data 'dimensions', that is, the number of values of each independent variable; and the type of the data, *e.g.*, character, integer, floating point, or binary.

Other criteria that should be considered are the following.

- Use a methodology that is extensible, so if the data standard needs to be revised or expanded, the methodology should be flexible enough to accommodate changes.
- Use or develop data standards that support both exchanging data *and* archiving data.
- Consider only data standards that are database-independent, so any data standard can be implemented in any commonly used database management system.

Finally, as described in the section on Obstacles below, *it is essential that data standards developed for the GTL Program be simple enough to be adopted easily by the research community.* The objective is to have a simple and useful data standard that will be used rather than a comprehensive, but overly complicated standard that is never used.

Data Standards: Description and Implementation

In general, a data standard can be **described** in more than one way, including by a text document, a file format with fixed fields and records, a modeling language diagram (*e.g.*, Unified Modeling Language), a markup language (which in turn may be described in different schema formats), or a script for creating tables in a database. Similarly, data standards can be **implemented** in a variety of formats and, in some cases, the description of the data standard is the same as the implementation (*e.g.*, a file format). Other implementations include markup languages, databases that mirror the data standard, and software tools for metadata entry that output the metadata in the format or markup language of the data standard. Distinguishing between data standard descriptions and implementations, however, is not as important as understanding how specifying a file format or markup language for a data standard leads to the development of compatible databases and minimizes duplication of effort in software development.

Software Tools to Support Data Standards

The use of data standards depends primarily on the availability of easy-to-use software tools to put metadata and data into the data standard format. Such tools include 'type and enter' tools, either Web-based or stand-alone, and conversion software to convert data taken from instruments/equipment into the data standard format [3, 4]. Development of these tools is essential for the success of data standards. Definition of the data standard provides the detail that software developers need to develop

- programs that transfer the content from the data standard file into display and analysis tools,
- Structured Query Language (SQL) scripts to create database tables, and
- software to enter the content of data standards files, *i.e.*, the metadata and data (or pointers to the data), into the database.

Common Obstacles to Developing and Using Data Standards

Data standards are boring. Defining data standards is far less interesting than designing an experiment or analyzing the results of an experiment. Given a choice between writing a report on data standards, or running another round of experiments, most researchers will opt to work in the lab.

Data standards are complicated. This may be the case, but presentation of data data standards and the software tools that implement data standards do not have to be complicated or hard to use. Where computer science issues are involved, and choices need to be made by experimentalists involved in defining data standards, experimentalists need to be educated, not overwhelmed by computer science jargon.

Data standards take a long time to develop. This is the case, but as the approach in the following section shows, we advocate developing data standards in stages, using existing data standards wherever possible, and collaborating with other data standards groups. In addition, we recommend defining a set of descriptive terms, *i.e.*, a 'controlled vocabulary' rather than a full-blown ontology to start with.

No one will use the data standards. Data standards do not or should not exist in a vacuum. Easy-to-use, well designed software tools are needed to encourage the use of data standards. In addition, format converters to bridge the gap between data being produced by equipment and proposed data standards need to be developed.

Approach

We already have compiled lists of data standards for general scientific data and for proteomics and transciptomics experiments [5,6]. In addition, a (UML) schema for high-level experiment and protocol metadata experiments [7] is available.

The remaining steps of Phase I are as follows:

- identify GTL researchers who are willing to participate in the review, development, and testing process;
- identify an initial set of data standards to define and develop based on existing or developing data standards and the highest priority needs (*e.g.*, sample preparation and generation) of the GTL Program;
- obtain sample data sets that can be used to test the data standards;
- talk to researchers who are using the data standards or who have reviewed them for use;
- define data standards for high-level experiment and protocol metadata;
- contact other groups (*e.g.*, MGED, NIST) developing data standards in order to facilitate the exchange of information;
- compare, feature by feature, existing data standards for the same types of experiments in order to determine the best practices used and where one data standard is better than another;
- determine how versioning and data access will be handled in the context of the data standards;
- prepare draft data standards for each type of GTL experiment by choosing an existing data standard, by combining existing data standards, or by developing a data standard from scratch if none exists;
- post the draft data standards on a Web site and notify participants that the data standards are ready for review;
- collect reviews and circulate them among participants;
- hold a review meeting to settle differences;
- post the first versions of the data standards.

Phase II will include the following steps:

- compile a list of equipment being used in existing GTL projects and the equipment vendors;
- obtain vendor-data formats;
- develop software tools to convert vendor-formatted information into the data standard developed in Phase I;
- develop information entry tools to capture high-level experiment and protocol metadata;
- document and write tutorials for the software tools and create installation software that is easy to use;
- train a small number of researchers or their staff members to use the software tools.

Note that the first two steps of Phase II may be taken during Phase I above.

Phase III will be to review and refine the data standards and the software tools and to add data standards for additional types of experiments. Phase IV will be to develop the SQL scripts to create database tables and software to enter the contents of the data standard files into a database.

Throughout all phases, it will be necessary to try to maintain backward compatibility with existing data standards. For example, the GTL Program may develop a data standard based on an existing one, but may revise or extend it. Though we anticipate actively promoting the data standards developed by the GTL Program, nevertheless, in order to maintain compatibility with segments of the scientific community using the non-GTL version of the data standard, it may be necessary to provide a mapping between data standards.

Phase V would be to create a data archive for the GTL Program, but would require significant more resources than are required for the data standards effort alone.

Appendix: Description of Information/Data Package and Collection

GTL experiments may be broadly characterized by organism, experimental technique, experimental conditions, *i.e.*, treatment or physical environment, and participating institution. We anticipate developing the requirements for an information/data package that initially will contain the following blocks of information and data:

- project information;
- sample preparation and generation, including media preparation and culture conditions (*e.g.*, pH, temperature, gas pressure, cell density at time of harvest, growth phase, *etc.*) and quality control/quality assurance procedures;

- information about a *set* of experiments, including institution and contact information, overall objective, experimental techniques used, reference publications, experimental factors varied, *etc.*;
- biological context, which in general terms (not all of which are applicable to GTL) would include organism (genus, species, sub species, strain, mutatation, *etc*.), location of the biological sample in the organism (organ, tissue, cellular location, *etc*.), gender, developmental stage (age, generation, *etc*.), diseased or normal state;
- environmental context, which may include location where organism was collected, environment in which organism is found, environmental conditions such as pH, aerobic or anerobic conditions, *etc.*

The blocks in the list above would occur once in the information/data package, and then for each experiment in the set of experiments, there would be blocks for,

- experiment design, including replicate information, operator information, equipment used, experiment factors or treatment, other experiment operating parameters;
- data from the experiment, including whether they are raw, processed, or interpreted data; any encoding used; the data types; how the data correlate with experiment factors (*e.g.*, time series, change of pH, nitrogen or oxygen levels, *etc*.), that is, the data format; if the data are processed or interpreted, what analysis protocol was used; links to external files if the data are not contained within the information/data package; and if the data are available in a file external to the information/data package, then the method used to compress the data and the method or software that can read the data file.

The data block is the most difficult to deal because of the possibly large size of the data set; because the data can be processed or interpreted, and because many experiments are performed in replicate. If the raw data set is very large, then a decision must be made whether to include it in the information/data package or only to include an external file reference to it. The external reference could be to an File Transfer Protocol (ftp) site (not desirable), or to a file that is combined with the information/data package to form an 'information/data collection'. Such a collection of files could be compressed using 'zip' or 'tar' software for transport or for archiving.

If the data are processed (*e.g.*, averaged or normalized) or interpreted, then enough information about the processing or methodology that led to the interpretation needs to be included in order to make the data reproducible. If the experiment was performed as a set of replicates (same experimental design and experimental conditions), then it is essential to provide information to associate replicates of data sets.
Descriptions of quality control measures or how results were normalized may be part of the experiment set level description or the experiment design level description.

The experiment design block will vary considerably from experimental technique to experimental technique, however, this is the place where we can use or adapt existing data standards for specific experimental techniques. One difficulty in using or adapting them is basic information such as names of researchers, contact information (*e.g.,* phone numbers and addresses), dates, publication information, measurement units, *etc.* are not represented in the same way from data standard to data standard. It will be necessary, therefore to use the best practices of existing data standards (not only data standards in the biological sciences) to cover these types of information. Specifying units of measurement is particularly troublesome because there is no widely used data standard for units, although the National Institute of Standards and Technology (NIST) has a working group on this topic [8], and the Systems Biology Markup Language (SBML) has a consistent way of representing units using a markup language [9].

References

[1] Microarray Gene Expression Data Society (2002). *Minimum Information About a Microarray Experiment – MIAME 1.1 Draft 6*. Available via the World Wide Web at http://www.mged.org/Workgroups/MIAME/miame_1.1.html.
[2] See Reference [6], and in particular, *Appendix A: Base64 Encoding*.
[3] Institute of Systems Biology (2003). *SASHIMI Software*. See the list of file converters for raw mass spec data (ReAdW, MassWolf, mzStar) described on the SASHIMI Software page: http://sashimi.sourceforge.net/software.html.
[4] Thermo Galactic (2002). *File Converters*. Information available via the World Wide Web at http://www.galactic.com/instruments/file_converter.htm.
[5] GTL Data Standards Working Group (2004). *Compilation of Data Standards for Proteomic and Transcriptomic Experiments.* Available via the World Wide Web at http://vimss.lbl.gov/~jsjacobsen/GTL/datastds.html.
[6] GTL Data Standards Working Group (2004). *Developing Standards for Data Generated by GTL Experiments*. Available via the World Wide Web at http://vimss.lbl.gov/~jsjacobsen/GTL/scidata.html.
[7] Jacobsen, J. (2003). *BioDB Schemas for Experiment and Protocol Metadata*. Available via the World Wide Web at http://vimss.lbl.gov/~jsjacobsen/BioDB/metadata.html. Note that this Web site is password protected (username=datastd, password=data).
[8] Dragoset, B. National Institute of Standards and Technology (2003). *Units Markup Language (UnitsML)*. Available via the World Wide Web at http://unitsml.nist.gov.
[9] Finney, A. and Hucka, M. (2003). *Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions*. See *Section 4.4 Unit Definitions*. Available via the World Wide Web at http://www.sbml.org/specifications/sbml-level-2/version-1/html/sbml-level-2.html.

Compilation of Data Standards for Proteomic and Transcriptomic Experimental Data

Genomes To Life Data Standards Working Group
Argonne National Laboratory | BIATECH | Harvard University | Howard Hughes Medical Institute Lawrence Berkeley National Laboratory | *Los Alamos National Laboratory* | *Oak Ridge National Laboratory Pacific Northwest National Laboratory University of California at Berkeley* | *University of California at San Diego*

Contents

Contributing Members

John Ambrosiano, ambro@lanl.gov
Computer and Computational Sciences
Los Alamos National Laboratory

Adam Arkin, *Chairperson*, APArkin@lbl.gov
Physical Biosciences Division
Lawrence Berkeley National Laboratory
Department of Bioengineering
University of California, Berkeley
Howard Hughes Medical Institute

Gyorgy Babnigg, GBabnigg@anl.gov
Biosciences Division
Argonne National Laboratory

Ed Frank, EFrank@mcs.anl.gov
Mathematics and Computer Science Division
Argonne National Laboratory

Al Geist, gst@ornl.gov
Computer Science and Mathematics Division
Oak Ridge National Laboratory

Carol Giometti, CSGiometti@anl.gov
Biosciences Division
Argonne National Laboratory

Janet Jacobsen, JSJacobsen@lbl.gov
Institute for Quantitative Biomedical Research
University of California, Berkeley

Nagiza Samatova, SamatovaN@ornl.gov
Computer Science and Mathematics Division
Oak Ridge National Laboratory

Nancy Slater, NASlater@lbl.gov
Physical Biosciences Division
Lawrence Berkeley National Laboratory

Ron Taylor, Ronald.Taylor@pnl.gov
Biological Sciences Division
Pacific Northwest National Laboratory

Other members of the GTL Data Standards Working Group are

- Eric Alm, Physical Biosciences Division, Lawrence Berkeley National Laboratory
- Eugene Kolker, BIATECH
- Bertram Ludaescher, San Diego Supercomputer Center, University of California, San Diego
- George Michaels, Biological Sciences, Pacific Northwest National Laboratory
- D. William Nguyen, Harvard Medical School, Harvard University
- Bahram Parvin, Computational Research Division, Lawrence Berkeley National Laboratory
- Ed Uberbacher, Life Sciences Division, Oak Ridge National Laboratory

Objective

The purpose of this review is to compile a list of data standards developed for, or applicable to, experimental methods used in proteomic and transcriptomic experiments. These experimental methods include, but are not limited to, flow cytometry, mass spectrometry, microarray, and microscopy. The underlying objective of this review is to initiate the definition and development of data standards. Adoption and use of data standards will facilitate data exchange and data *integration*, so that databases can be built that contain information not just about a single type of experiment, but that associate genes and proteins with a variety of experimental data and data annotations.

Scope

This compilation is limited to experimental methods used in proteomic and transcriptomic experiments primarily because these methods will be widely used throughout the Genomes To Life Program, and also because of current efforts underway to define data standards for those methods. This compilation does not include data standards for genomic data or data standards applicable to sample preparation and generation.

This document describes both proposed data standards as well as data standards implemented as file formats, XML schemas, or databases. Open source or freely available software packages developed for data entry, retrieval, display, and analysis have been included as well.

The following section describes data standards initiatives for analytical data (*e.g.*, gas chromatography and mass spectrometry), microarray data, and mass spectrometry data in the context of proteomics experiments. The section on data standards initiatives is followed by sections describing how some data standards have been implemented as file formats, markup languages, databases, and/or software environments.

Data Standards Initiatives

Several data standards committees and collaborative ventures are in the process of developing recommendations for data standards for proteomic and transcriptomic experiments. One of the most active groups developing standards for **microarray gene expression experiments** is the Microarray Gene Expression Group (MGED) Society. MGED was founded as a grass roots movement at the end of 1999 by an international group of microarray developers and users who realized the need for standardizing annotation information in order to "facilitate the sharing of microarray data generated by functional genomics and proteomics experiments". MGED has been responsible for developing the data standard, MIAME (Minimum Information About a Microarray Experiment), as well as implementations of MIAME (MAGE-ML, MAGE-OM, and MAGE-stk), all of which are described in Appendix A.

The goal of the Proteomics Standards Initiative (PSI) is to set "community standards for data representation in proteomics to facilitate data comparison, exchange, and verification". PSI was founded in April 2002 by the Human Proteome Organization, an international organization that includes universities, government agencies, and industry participants. Currently, PSI has three working groups - protein-protein interactions, mass spectrometry, and general proteomics format - and plans to develop standards for the first two types of data and to participate in developing standards for the third.

PSI's Web page notes the existence of several well-established databases for **protein-protein interaction data** (*e.g.,* BIND, DIP, MINT, and MIPS) as well as the need to synchronize the core data provided by public protein interaction databases. PSI's goal for **mass spectrometry experiments** is to "develop a standard representation of experimental spectra in the context of the experimental setup and the analyzed system". With respect to **general proteomics format**, PSI intends to endorse/co-develop an existing model rather than being a new effort. PSI's Web page provides links to HUP-ML (Human Proteome Markup Language) and PEDRo (Proteomics Experiment Data Repository), both of which are described in Appendix A.

In addition, there are government agencies and industry organizations developing data standards for **analytical data**, **units of measurement**, *etc.*, some of which are listed in Appendix B. Most notable of these efforts is the Analytical Data Exchange effort of the American Society for Testing and Materials (ASTM) Subcommittee on Analytical Data Management (E13.15). The mission of this subcommittee is to define "standards for representing, managing, and interchanging analytical chemistry data including the implementation of technique specific information and application to instrument data interfaces". The subcommittee met in March 2003 to review two data standards for analytical data: GAML and SpectroML. At that meeting, it was proposed that a new data exchange standard, AnIML (Analytical Information Markup Language), be developed based on the best features of GAML and SpectroML. AnIML is of interest because one of its applications is to mass spectrometry data. Because annotated peak lists (*i.e.*, peak lists together with protein identifications) are particularly important in proteomics-related mass spectrometry experiments, AnIML is not by itself sufficient as a data standard for such experiments.

Data Standards Reviewed

Listed below are the data standards, implementations, and software packages reviewed in this document. Note that this list is not exhaustive and that some of these data standards are still under active development and not yet in widespread use. Each standard in the table below is described in more detail in Appendix A. The following section briefly describes the data standards according to whether or how each has been implemented.

BASE	BioArray Software Environment
CytometryML	Cytometry Markup Language
ExperiBase	A Software Platform to Support Modern Experimental Biology
FCS	Flow Cytometry Standard
HUP-ML	Human Proteome Markup Language
MAGE-ML	Microarray Gene Expression Markup Language
MAGE-OM	Microarray Gene Expression Object Model
MIAME	Minimum Information About Microarray Experiments
mzXML	File Format Standard for the Representation of Mass Spectrometry
PEDRo	Proteomics Experiment Data Repository
PEML	Proteomics Experiment Markup Language
PRIME	Proteome Research Information Management Environment
SBEAMS	Systems Biology Experiment Analysis Management System

Table 1. Data Standards Reviewed

Implementations of Data Standards

This section describes some of the methods for implementing the data standards in Table 1. In general, a data standard for an experiment is a prescription for the types of data and information that must be included to fully describe the experimental conditions, the data collected, and usually the analysis of the data. A data standard can be implemented in different ways. Using a structured file format or developing a markup language for the data standard provides a machine readable form that can then be used to develop software tools for data storage, display, or analysis. In some of the examples below, software packages have been developed based on file formats or markup languages, so that a given data standard may have more than one implementation. Distinguishing between implementations is not what is important, but rather understanding how specifying a file format or markup language for a data standard leads to the development of databases that are compatible with one another and software tools that can focus on a single set of database tables or objects instead of multiple versions.

File Format

FCS is a data file standard for flow cytometry data. The first version of FCS was published in 1984. Recently, FCS has been challenged by a new data standard, CytometryML described in the next subsection.

XML Implementations

XML, the eXtensible Markup Language, provides "syntax for expressing structure in data". It is extensible because it does not use pre-defined tags to describe the content of a document or file; instead the user creates his/her own tags to describe data elements. Several of the data standards reviewed in this document have XML implementations: CytometryML (Cytometry Markup Language) and PEML (Proteomics Experiment Markup Language) have been implemented as XML schemas; HUP-ML (Human Proteome Markup Language) and MAGE-ML (MicroArray and Gene Expression Markup Language) have been implemented as DTDs (Document Type Definition).

Database Implementations

A database implementation depends on the development of a data model, which can be described using an entity-relationship diagram. A database usually is created using a script containing SQL "create table..." statements. SQL scripts are available for the following database implementations: BASE (BioArray Software Environment), MAGE-OM (MicroArray and Gene Expression Object Model), and PEDRo (Proteomics Experiment Data Repository).

Software Tools/Packages

Several of the data standards have software tools or packages associated with them. In most cases, the software consists of a user interface to allow the user to enter information about the experiment (conditions, protocols, equipment, *etc.*) and to upload data. The information and data are then loaded into a database.

Some of the software packages also include display and analysis tools as well as interfaces to public protein databases.

Public Data Repositories

Public data repositories are included in this document because the database implementation and user interface to the data repository represent a relatively stable state of development, as well as public acceptance, of the underlying data standard. ArrayExpress is a public repository for microarray data. The first submissions to ArrayExpress were made in 200_. ArrayExpress currently (October 2003) holds 56 experiments, 82 arrays, and 393 protocols from several different research groups.

Summary of Data Standards by Implementation

The following table summarizes data standards by the way in which they have been implemented.

type of implementation	**data standard/implementation**
file format	FCS
XML	CytometryML, HUP-ML, MAGE-ML, mzXML, PEML
database	BASE, MAGE-OM, PEDRo, SBEAMS
software tools	BASE, ExperiBase, HUP-ML Editor, MIAMExpress, PEDRoDC, PRIME, SBEAMS
public data repository	ArrayExpress (see MAGE-OM)

Table 2. Summary of Data Standards by Implementation

Data Standards and Implementations Listed by Type of Analytical Method

Some of the data standards and software implementations in the table below are specific to one type of experimental method, whereas others cover more than one (*e.g.*, ExperiBase).

type of analytical method	data standard(s) / implementation(s)
flow cytometry	CytometryML
	ExperiBase (includes flow cytometry)
	FCS
mass spectrometry	ExperiBase (includes mass spec)
	mzXML
	PEDRo / PEDRoDC / PEML
	PRIME
microscopy	ExperiBase (includes microscopy)
	FMAS
	OME
microarray	BASE
	ExperiBase (includes microarray)
	MAGE-ML / MIAME / MIAMExpress
2D gels	ExperiBase (includes 2D gels)
	HUP-ML (includes 2D gels)
	PEDRo / PEDRoDC / PEML (include 2D gels)
	PRIME (includes 2D gels)

Table 3. Data Standards and Implementation Listed by Type of Analytical Method

Experiment Metadata

Experiment metadata, that is, information *about* an experiment, are necessary to provide the context for the experiment. Without knowing the details of how two or more experiments were conducted, it is impossible to compare their results. Many of the data standards discussed above include schema for experiment metadata and in some cases, protocol metatdata. Some data standards, *e.g.,* MIAME, were developed specifically to capture information about experiments and protocols.

Not surprisingly, however, metadata schemas differ in what information about experiments and protocols they include. Moreover, basic information such as names of researchers, contact information (*e.g.,* phone numbers and addresses), dates, publication information, measurement units, *etc.* are not represented in the same way from schema to schema. If the schemas for different types of experimental methods were combined, the parts of the schemas covering the basic information would need to be normalized, and the best practices of each schema adopted and applied throughout the combined schemas.

As part of the development of BioDB, the database for the DARPA BioSPICE Program, schemas were developed for experiment and protocol metadata. These schemas, which are based on MIAME and other schemas, can be viewed at http://vimss.lbl.gov/~jsjacobsen/BioDB/metadata.html. (Note that this page is password protected. Send email to <JSJacobsen@lbl.gov> to request the username and password.)

Summary and Discussion

This document presents a compilation and review of data standards applicable to proteomic and transcriptomic experiments. The information was compiled primarily through Internet searches and publications available on-line. Most of the descriptions of data standards or their implementations in Appendix A are direct quotes from on-line source material. In spite of the fact that this document is not a *critical* review of proposed or existing data standards, it nevertheless is a good starting point for discussions on data standards for proteomic and transcriptomic experimental data.

Particularly striking about the data standards, *etc.,* reviewed herein is that with the exception of the Flow Cytometry Standard (FCS) originally published in 1984, none of the data standards or implementations is over four years old, and several are less than one year old. This is an area that will be undergoing rapid development and refinements during the next few years; however, there is widespread agreement on the need for data standards.

Not covered in this compilation are data standards for sample preparation and generation. In a research program such as the Genomes To Life Program, samples will be prepared at several labs and shipped to other labs and universities for further testing and analysis. It is critical, therefore, to define data standards for sample preparation in order to capture the detailed information essential for replicating experiments and tracking down anomalous results, and useful for future modeling efforts. Data standards for sample preparation and generation include media preparation and culture conditions (*e.g.*, pH, temperature, gas pressure, cell density at time of harvest, growth phase, *etc.*) and quality control/quality assurance procedures and results. Many of the data standards covered in this compilation include sample origin and/or preparation, but there are differences in how the information is represented and in what detail.

The next steps that need to be taken with respect to this compilation, however, are as follows.

1. Obtain feedback on this document in terms of its completeness and accuracy.
2. Solicit information from researchers using existing data standards and software tools to find out about their experiences - both good and bad.
3. Request that domain experts evaluate comparable data standards.

Taking the steps above should provide the basis to make decisions on which data standards and software implementations to test further, to adopt, to adapt, or to invest resources to continue their development.

List of Tables

Acknowledgement

We would like to thank Frank Olken of the Scientific Data Management Research Group, Computational Research Division, at Lawrence Berkeley National Laboratory for his review of and constructive comments about this Web document.

Request for Feedback

Please send your comments on this document to the members of the GTL Data Standards Working Group at gtl-workinggroup@vimss.lbl.gov.

Date of Last Revision: January 26, 2004

Appendix A: Data Standards for Proteomics and Transcriptomic Experiments

This appendix contains a section for each of the following data standards/implementations of data standards:

- BASE - BioArray Software Environment
- CytometryML - Cytometry Markup Language
- ExperiBase - A Software Platform to Support Modern Experimental Biology
- FCS - Flow Cytometry Standard
- HUP-ML - Human Proteome Markup Language
- MAGE-ML - Microarray Gene Expression Markup Language

- MAGE-OM - Microarray Gene Expression Object Model
- MIAME (and MIAMExpress) - Minimum Information About Microarray Experiments
- mzXML(and SASHIMI) - File Format Standard for the Representation of Mass Spectrometry Data
- PEDRo - Proteomics Experiment Data Repository
- PEDRoDC - PEDRo Data Collator
- PEML - Proteomics Experiment Markup Language
- PRIME - Proteome Research Information Management Environment
- SBEAMS - Systems Biology Experiment Analysis Management System

BASE - BioArray Software Environment

URL: http://base.thep.lu.se/

Goal: to develop "a comprehensive free web-based database solution for the massive amounts of data generated by microarray analysis"

Description: "BASE is a comprehensive database server to manage the massive amounts of data generated by microarray analysis. In short, it manages biomaterial information, raw data and images, and provides integrated and plug-in-able normalization, data viewing and analysis tools. Additionally, for labs that make their own in-house arrays or for labs that wish to track probe information, the system also has array production, LIMS features which can be integrated with the data analysis. The organization and interface of BASE was designed to closely follow the natural work-flow of the microarray biologist, and is compatible with most types of array platforms and datatypes (*e.g.,* cDNA/oligos spotted on any substrate, Affymetrix, CGH on arrays, *etc.*)."

Developer(s): Department of Theoretical Physics, Lund University, Sweden

Status:

- First released in May 2002.
- BASE 1.2.9 released October 14, 2003.

Availability:

- Available under a GNU General Public License (GPL).
- BASE 1.2 is available from SourceForge.net's CVS server.

- BASE 1.3 is maintained by Alan Shields at the Oklahoma Medical Research Foundation (OMRF) Microarray Facility using the arch revision control system.

Implementation:

- BASE is a software environment that was developed "using free software: Linux OS, MySQL database, Apache webserver, C++/Javascript/PHP languages".
- BASE is meant to be installed on a local server (running Linux) and accessed via a Web browser (running on a Mac or on a PC running Linux or Windows).
- As noted above, the database implementation is for a MySQL database. The SQL to create the BASE MySQL database is available from the SourceForge.net CVS repository.

Publication(s):
Lao H. Saal, Carl Troein, Johan Vallon-Christersson, Sofia Gruvberger, Åke Borg and Carsten Peterson. BioArray Software Environment: A Platform for Comprehensive Management and Analysis of Microarray Data. *Genome Biology* **3**(8), software0003.1-0003.6 (2002).

Comments:

- The size of the user base is unknown, but 1,300 people have downloaded BASE 1.0.x or 1.2 since March 2003.
- BASE is MIAME compliant and will support data export in MAGE-ML.
- BASE 2, which is a rewrite of BASE 1.x, is in progress. The technical specification and feature list for BASE 2 may be found at http://www.thep.lu.se/~nicklas/base2/

CytometryML - Cytometry Markup Language

URL:
http://www.newportinstruments.com/cytometryml/cytometryml.html

Goal: "to produce a set of XML schemas to define cytometry data"; proposed as a replacement for the Flow Cytometry Standard (FCS)

Description: "Cytometry Markup Language, CytometryML, is a proposed new analytical cytology data standard. CytometryML is a set of XML schemas for encoding both flow cytometry and digital microscopy text based data types. CytometryML schemas reference both DICOM (Digital Imaging and Communications in Medicine) codes and FCS keywords."

Developer(s): XML_Med, a Division Newport Instruments, San Diego, California

Status:
Description of CytometryML published in 2003 (several prior publications advocating the use of an XML description for cytometry data rather than FCS).

Availability:
CytometryML schemas and documents may be downloaded from http://www.newportinstruments.com/cytometryml/cytometryml.html

Implementation:
Implemented as a markup language, but so far no reports of actual use found.

Publication(s):

- R.C. Leif, S.H. Leif, S.B. Leif. CytometryML, An XML Format based on DICOM for Analytical Cytology Data, *Cytometry* **54A**(1), 56-65 (2003).
- R.C. Leif, S.H. Leif, S.B. Leif. CytometryML, a Markup Language for Analytical Cytology, to appear in *SPIE Proceedings* **4962** (2003).
- R.C. Leif and S.B. Leif. A DICOM Compatible Format for Analytical Cytology Data, that can be Expressed in XML, in *Optical Diagnostics of Living Cells IV*, D. L. Farkas and R. C. Leif (eds), *SPIE Proceedings* **4260**, 238-48 (2001).

Comment:
CytometryML imports data types from the Digital Imaging and Communications in Medicine (DICOM) standard, FCS, and MathML.

ExperiBase

URL: http://schiele.mit.edu:8080/index.html

Goal: "to develop a new informatics platform to handle... gel electrophoresis, microarrays, fluorescence activated cell sorting, mass spectrometry, and microscopy within a single coherent set of information object definitions"

Description: ExperiBase "can store and query data generated by the leading experimental protocols used in biology within a single database. ExperiBase also has provisions to store derived data from analysis as a part of an expanded definition of the information object. Transport of the raw data and analytical results between ExperiBase and external analysis packages uses web-based network technologies and XML representation of the data itself. The information object model is used to define the form of the XML data document. Import and export of data in spreadsheet format is also supported. ExperiBase has been ported to three leading database platforms: Oracle, DB2 and Informix. There are no platform-specific dependencies." ExperiBase provides support for the following types of experiments/data: FACS, flow cytometry, gel electrophoresis (Western blot, 1D gel, 2D gel), mass spectrometry, microarray, and microscopy.

Developer(s): Developed by researchers in the Department of Mechanical Engineering and the Division of Biological Engineering at the Massachusetts Institute of Technology, Cambridge, Massachusetts, with support from DARPA (Defense Advanced Research Projects Agency) and DOE through PNL (Pacific Northwest National Laboratories).

Status:

- The XML schema for ExperiBase was circulated among members of the I3C (Interoperable Informatics Infrastructure Consortium) Life Science Object Ontologies group in September 2003 and is on the agenda for the I3C Technical Meeting at the end of October 2003.

- ExperiBase has been implemented using several database management systems. See the **Implementation** notes below.

Availability:
The ExperiBase XML schema (v0.1) and documentation may be downloaded from http://schiele.mit.edu:8080/index.html.

Implementation:

- As noted above, an XML schema for ExperiBase is available.
- ExperiBase 1.0 (gel chromatography only) was implemented as an Informix database. This version has a Web-based "viewer" that lists all experiments, protocols, cells, chemicals, and process methods. Information/documentation for this version may be found at http://schiele.mit.edu:8080/research/ExperiBase_introduction_files/v3_document.htm.
- ExperiBase 2.01 Explorer was implemented as a DB2 (v8.1) database. There is a demo page for this implementation at http://schiele.mit.edu:8080/ExperiBase/index.htm.
- ExperiBase 2.02 Explorer was implemented as an Oracle 9i (rel. 2) database. There is a demo page for this implementation at http://schiele.mit.edu:8080/ExperibaseOracle/index.htm.

Publication(s): No journal publications or conference papers to date.

Comments:

- ExperiBase uses, or is based, on published ontologies such as MAGE for microarrays, PEDRo for mass spec, OME for optical microscopy, and CytometryML for flow cytometry. The developers of ExperiBase have developed their own ontologies for gel electrophoresis.

- Demonstration of ExperiBase Explorer uses data from FACS experiments (MIT, Facsimile datasets from the Australian National University, and FlowJo datasets from www.flowjo.com), microarray experiments (Stanford Microarray Database), and Western blot experiments (MIT).

FCS - Flow Cytometry Standard

URL: http://www.isac-net.org/

Goal: "to facilitate the development of software for reading and writing flow cytometry data files in a standardized format"

Description: "The flow cytometry data file standard provides the specifications needed to completely describe flow cytometry data sets within the confines of the file containing the experimental data... The principal goal of the Standard is to provide a uniform file format allowing files created by one type of acquisition hardware and software to be analyzed by another type... The FCS structure requires that each data set in a file contains three segments: HEADER, TEXT and DATA, with an optional ANALYSIS segment."

Developer(s): Developed by ISAC (International Society for Analytical Cytology).

Status:

- The original FCS standard was published in 1984 as FCS version 1.0 and was amended in 1990 as FCS version 2.0.
- FCS3.0 was released in 1998.

Availability:
FCS3.0 may be downloaded from http://www.isac-net.org/. (Click on the "Links & Resources" link and then on the "FCS 3.0" link.)

Implementation:

- FCS is a prescribed *file format*.

- There are freeware/shareware cytometry programs listed on the ISAC Web site. (Click on the "Links & Resources" link and then on the "Software" link.) These programs read flow cytometry files and display and analyze data.

Publication(s):

- (FCS 1.0) Murphy RF and Chused TM. A proposal for a flow cytometric data file standard. *Cytometry* **5**, 553-555 (1984).
- (FCS 2.0) Dean PN, Bagwell CB, Lindmo T, Murphy RF, and Salzman GC. Data File Standard for Flow Cytometry. *Cytometry* **11**(3), 323-32 (1990).

Comments:

- The FCS file format has four sections: HEADER, TEXT, DATA, and ANALYSIS.
- The FCS 3.0 document contains an appendix title, Proposed API for reading and writing FCS files. The date on the API description is May 2000, however.

HUP-ML - Human Proteome Markup Language

URL:

- http://www1.biz.biglobe.ne.jp/~jhupo/HUP-ML/hup-ml.htm (newer version)
- http://www1.biz.biglobe.ne.jp/~jhupo/HUP-ML/hup-ml.html (older version)

Goal: to develop a standard for sharing information on sample preparation and experimental conditions for proteome experiments and analysis

Description: "HUP-ML (Human Proteome Markup Language) is a XML based and proteomics-oriented markup language for exchanging proteome data between researchers to accelerate their collaboration. HUP-ML contains not only the details of methodology and experimental conditions, but also the results of proteome analysis."

Developer(s): Developed by the Proteomic Research Center, NEC Corporation, Japan

Status:

- HUP-ML was proposed at the AOHUPO XML Workshop in December 2002. The current version of HUP-ML covers 2D electrophoresis experiments; "liquid chromatography profiles will be adopted next".
- Version 0.43 (beta) of the HUP-ML Editor was released in January 2003.
- Version 0.80 of the HUP-ML Editor is the latest release.

Implementation:

- HUP-ML is available as a DTD (Document Type Definition).
- The HUP-ML Editor is a graphical user interface that can "create and edit HUP-ML formatted data". The HUP-ML Editor has the following capabilities:
  - "template functions to avoid re-entry of the same information",
  - "clickable 2D gel image viewer",
  - "viewer of public database through the accession number acquired by protein identification",
  - "protein data importing function to merge related data from public data base",
  - "MS raw data importing and charting function".

Availability:

- Version 0.08 of the DTD for HUP-ML and version 0.80 of the HUP-ML Editor can be downloaded from http://www1.biz.biglobe.ne.jp/~jhupo/HUP-ML/hup-ml.htm.
- (Version 0.43 of the DTD for HUP-ML can be downloaded from http://www1.biz.biglobe.ne.jp/~jhupo/HUP-ML/hup-ml.dtd.)

- (Version 0.43 (beta) of the HUP-ML Editor may be downloaded from http://www1.biz.biglobe.ne.jp/~jhupo/HUP-ML/hup-ml-editor.html.)

Publication(s):
K. Kamijo *et al.* A Proposition of XML Format for Proteomics Database. Proceedings Of the 18th International CODATA Conference, p.50 (2002).

Comments on the HUP-ML Editor:

- The HUP-ML Editor requires the Microsoft Windows operating system.
- The user's guide for the HUP-ML Editor is in Japanese.
- *Need to check the License Agreement in the installation package.*

MAGE-ML - Microarray Gene Expression - Markup Language
MAGE-OM - Microarray Gene Expression - Object Model

URL: There are links to MAGE-ML and MAGE-OM from the MAGE home page: http://www.mged.org/Workgroups/MAGE/mage.html.

Goals:

- MAGE-ML: to establish a data exchange format
- MAGE-OM: to establish a data exchange model

Description(s): MAGE-ML and MAGE-OM are standards for representing microarray expression data. MAGE-OM is an object model that has been modeled using UML (Unified Modeling Language) MAGE-ML is a data exchange format that has been implemented using XML. MAGE-ML has been derived from MAGE-OM. MAGE-OM includes packages for analyses, array design, bioassays, "bioevents", biomaterials, experiments, measurements, and protocols. See the Introduction to MAGE-ML and MAGE-OM at http://www.mged.org/Workgroups/MAGE/introduction.

html for more information on MAGE-ML and MAGE-OM.

Developer(s): MAGE-ML and MAGE-OM were developed by the MGED MAGE group in collaboration with the OMG (Object Management Group) and other collaborators, including Rosetta Inpharmatics.

Status:

- The first version of MAGE-ML was finalized in January 2002. Version 1.1 was approved???
- The first version of MAGE-OM was finalized in September 2002. Version 1.1 was approved in May 2003.
- The first release of MAGE Java was in February 2002, and the first release of MAGE Perl was in September 2002. MAGE Java and MAGE Perl are part of MAGE-stk, the MAGE Software Toolkit.

Implementation:

- MAGE-ML has been implemented using XML. There is a DTD (Document Type Definition) available for MAGE-ML.
- MAGE-OM has been modeling using UML.
- MAGE-stk "is a collection of Open Source packages that implement the MAGE Object Model in various programming languages". The current releases include applications written in Perl and Java.
- A relational schema has been generated from MAGE-OM and has been used to create ArrayExpress, a public data repository for microarray data. ArrayExpress runs Oracle, but no Oracle-specific features have been used. The database scripts for creating the database tables are available from the EBI ftp site, file **MAGE-RS.tab**. ArrayExpress currently (October 2003) holds 56 experiments, 82 arrays, and 393 protocols from several different research groups. Data can be submitted to ArrayExpress using either MIAMExpress or as MAGE-ML formatted files.

Availability:

- MAGE-ML documents can be viewed or downloaded from links at http://www.mged.org/Workgroups/MAGE/mage-ml.html.
- MAGE-OM documents can be viewed or downloaded from links at http://www.mged.org/Workgroups/MAGE/mage-om.html.
- Packages available from http://sourceforge.net/projects/mged/ include
    - example MAGE-ML files,
    - a MAGE Java API,
    - a MAGE Perl API,
    - the OMG model (MAGE-OM),
    - and presentations on MAGE-OM, MAGEstk, and the MGED ontology.
- MGED **also** has an easier-to-navigate home page for the SourceForge site that has downloads for MAGE-ML, MAGE-OM, and MAGE-stk from both OMG and SourceForge.

Publication(s):
PT Spellman *et al.* Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol.* **3**(9):RESEARCH0046 (2002).

Comments:

- A good starting point for understanding the connection and differences between MAGE-ML and MAGE-OM is the MAGE Introduction.
- See also the section on MIAME (and MIAMExpress).

MIAME (and MIAMExpress) - Minimum Information About Microarray Experiments

URL:
http://www.mged.org/Workgroups/MIAME/miame.html

Goal: "to outline the minimum information required to unambiguously interpret microarray data" and "to guide the development of microarray databases and data management software"

Description: "MIAME aims to outline the minimum information required to unambiguously interpret microarray data and to subsequently allow independent verification of this data at a later stage if required. MIAME is not a dogma for microarray experiments to follow, but just a set of guidelines. This set of guidelines will then assist with the development of microarray repositories and data analysis tools... Although MIAME concentrates on the content of the information and should not be confused with a data format, it also tries to provide a conceptual structure for microarray experiment descriptions."

Developer(s): Developed by the MIAME working group of the Microarray Gene Expression Data (MGED) Society.

Status:

- Version 1.1 (Draft 6) of MIAME was released April 2002.
- MIAMExpress1.0 was released January 2003; MIAMExpress1.5 was released October 2003.

Implementation:

- MIAMExpress is "a MIAME compliant microarray data submission tool". The home page for MIAMExpress is http://www.ebi.ac.uk/miamexpress/. MIAMExpress is written in using Perl CGI scripts and uses MySQL as its database management system. Scripts to create MySQL tables are part of the MIAMExpress installation package.
- *See also MAGE-ML and MAGE-OM.*

Availability:

- Version 1.1 (Draft 6) of MIAME can be found at http://www.mged.org/Workgroups/MIAME/miame_1.1.html.

- MIAMExpress can be downloaded from http://sourceforge.net/projects/miamexpress/. MIAMExpress is platform-independent

Publication(s):
A. Brazma, *et al.* Minimum information about a microarray experiment (MIAME) - toward standards for microarray data. *Nature Genetics*, **29** (2001)

Comments:

- There is no user's guide for MIAMExpress other than the HTML help pages accessible when MIAMExpress is running.
- See http://www.mged.org/Workgroups/MIAME/miame_software.html for a list of "possibly MIAME compliant software".

mzXML (and SASHIMI) - File Format Standard for the Representation of Mass Spectrometry Data

URL: http://sashimi.sourceforge.net/

Alternate names: *Several different names for the file format appear on the SASHIMI SourceForge Web site: mzXML, ms_xml, MsXML, and MSXML.*

Goal(s): The goal of mzXML is to develop an open standard, XML-based file format for mass spectrometry data. The goal of the SASHIMI project is "to provide the scientific community with free open source software tools for the downstream analysis of mass spectrometric data."

Description(s): "To address the difficulties presented by the introduction of a new mass spectrometer into a pre-existing data analysis framework, we developed an XML based common file format for MS data. The adoption of an open standard will provide programmers with an easy way to access this kind of information, thus facilitating development and distribution of software in this field. Additionally, the use of an architecture and operating system independent representation will ease the exchange of datasets between collaborators and ultimately allow for the creation of public data repositories."

Developer(s): Proteomics Group, Institute for Systems Biology (ISB), Seattle, Washington

Status:

- mzXML was first publicly available in April 2003(?).
- The SASHIMI software tools (see http://sashimi.sourceforge.net/software.html) were added to the CVS repository starting in April 2003.

Availability:

- mzXML is available for downloading from SASHIMI's SourceForge CVS repository at http://cvs.sourceforge.net/viewcvs.py/sashimi/ms_xml/.
- The SASHIMI software tools are available for downloading at http://cvs.sourceforge.net/viewcvs.py/sashimi/. Each program is briefly described at http://sashimi.sourceforge.net/software.html. The Web page has the following disclaimer: "[These] are development versions and might (will) cause problems by compiling and or use. If you want to use them anyway do so at your own risk."
- Insilicos Viewer, a free viewer for mass spectrometry proteomics data that can read mzXML, is available from a company called Insilicos. A beta (object code) version of Insilicos Viewer, which runs on Windows 2000/XP can be downloaded from http://www.insilicos.com/products.html. The software is free, but requires agreeing to a "Beta Software End-User License Agreement".

Implementation:

- The mzXML file format was implemented in XML.

- The software tools in the SASHIMI project are a mix of C, C++, CGI, and Java.

Publication(s):
PGA Pedrioli, *et al.*, Creation of an open standard file format for the representation of MS data. 51st ASMS Conference, June 8-12, 2003, Montreal, Canada.

Comments:

- mzXML indexes the position of each scan using a "scan" element and uses base64 encoding for the peak data. There is an mzXML Data Repository that has sample mass spectrometry data files in both the manufacturer's format (ThermoFinnigan LCQ and Micromass Q-TOR Ultima) and in mzXML. The mzXML files illustrate how the scan indexing works and the use of the base64 encoding for the peaks.
- The emphasis in mzXML is on the mass spectrometry data, but it includes some metadata to describe the hardware (instrument attributes such as manufacturer, model, and type of mass spec) and software (attributes such as type, name, and version).
- All of the software in the SASHIMI CVS repository are described as development versions with a warning to use "at your own risk".
- Insilicos Viewer is free, but not open source.

PEDRo - Proteomics Experiment Data Repository
PEDRoDC - PEDRo Data Collator
PEML - Proteomics Experiment Markup Language

URL: http://pedro.man.ac.uk/

Goal: to develop a "standard representation of both the methods used and the data generated in proteomics experiments"

Descriptions:

- PEDRo is a model that describes "the data that are required to be captured from a proteomics

experiment (both results and metadata)". The PEDRo model covers sample generation and processing, mass spectrometry experiments and results (peak lists), and the results of performing a database search to identify proteins.
- PEML is an "XML Schema representation of the PEDRo model... for use as a data interchange format".
- PEDRoDC is a "PEDRo-compliant, Java-based data entry tool... [that] collates the data and metadata from an experiment into a single XML file... for submission to a PEDRo-compliant repository".

Developer(s): The development of PEDRo, PEDRoDC, and PEML was funded by the COGEME project of the BBSRC's Investigating Gene Function Initiative and the E-Science North West Centre.

Status:

- The description of PEDRo was first published in March 2003.
- PEDRo v1.3 was released in October 2003.

Implementation:

- A UML schema for the PEDRo model is available as a UML class diagram.
- An SQL implementation of PEDRo, *i.e.,* an SQL script to create database tables, also is available.
- PEML is available as an XML schema.
- PEDRoDC is a Java-based data entry tool "collates the data and metadata from an experiment into a single XML file".

Availability:

- The PEDRo UML class diagram may be viewed at http://pedro.man.ac.uk/model.shtml.
- The SQL script to create database tables is available at http://pedro.man.ac.uk/files/PEDRoCreateTables.sql.

- The PEML XML schema is available at http://pedro.man.ac.uk/files/PEDRoSchema.xsd.
- PEDRo is freely available, but requires registration (http://pedrodownload.man.ac.uk/) prior to downloading it. The PEDRo download includes a three-page license agreement.
- PEDRoDC requires JDK 1.4 and has been tested only on a Windows 2000 platform.

Publication(s):
CF Taylor *et al.*, A systematic approach to modeling, capturing, and disseminating proteomics experimental data. *Nature Biotechnology*, **21**(21), 247-254 (2003).

Comments:

- PEDRo is an "explicit model of data and metadata" in contrast to the object model approach taken by MAGE.
- The PEDRo schema uses a parameters file for the machine-generated parameters list.
- The authors note that some quality control measures/indicators for protein identification many need to be added to the schema.

PRIME - Proteome Research Information Management Environment

URL(s):

- http://prime.proteome.med.umich.edu/index.html
- http://www.proteome.med.umich.edu/andrewslab/projects/

Goal: to develop an information management environment for proteome research

Description: "PRIME is an acronym for Proteome Research Information Management Environment. This software is built on a distributed open architecture enterprise platform. It is used in information management, tracking, and analysis for proteomics research. PRIME has several major components. These include a laboratory information management system (LIMS), viewers for 2D Gel images and mass spectrometry plots, a streaming engine for real time data input from state of the art spectrometers, including both MS and MS/MS data, an automated protein database search engine for both Peptide Mass Fingerprint and MS/MS peptide fragment searches, and a data discovery toolkit. PRIME stores the significant experimental data, and search result data in a relational database for use in data mining. PRIME is highly scalable, secure, and extendable due to its distributed architecture."

Developer(s): Andrews Research Laboratory, University of Michigan, Ann Arbor, Michigan (PRIME also is part of the Michigan Proteome Consortium at the University of Michigan)

Status: An alpha version of PRIME was made publicly available in February 2003.

Availability:
The (alpha) version 0.0 of PRIME may be downloaded from http://141.211.141.216/prime/index.htm, but note that the download is available ONLY to users of Internet Explorer.

Implementation:

- The software architecture is described as "based on a structure similar to the Java 2 Platform, Enterprise Edition".
- Oracle 9i was used as the database management system.

Publication(s):
Ulintz, P.J., Bly, M.J., Hurley, M.C., Haynes, H.A., and Andrews, P.C., PRIME: A management and data mining system implemented for E. coli membrane protein analysis. 4th Siena 2D Electrophoresis Meeting, Siena, Italy, September 2000.

Comments:

- The description of PRIME on the Web site is interesting, but there is no database schema available on the Web site.
- ONLY Internet Explorer users may access the Web site for the alpha version of PRIME.
- Some experimental data have been entered into PRIME, and there is a demo version available for browsing. The demo pages have links to summary pages (*e.g.*, progress reports, MS summaries, MS/MS summaries, spot stats summaries, and gel summaries). Some of those summary pages have information or data, and some did not.

SBEAMS - Systems Biology Experiment Analysis Management System

URL: http://www.sbeams.org/

Goal: "to support the data being generated by local microarray, proteomics, immunohistochemistry, and other experiments"

Description: SBEAMS is a "a framework for collecting, storing, and accessing data produced by a wide variety of experiments. [SBEAMS] provides a customizable framework to meet the needs of modern systems biology research. It is composed of a unified state-of-the-art relational database management system (RDBMS) back end, a collection of tools to store, manage, and query experiment information and results in the RDBMS, a Web front end for querying the database and providing integrated access to remote data sources, and an interface to existing programs for clustering and other analysis.... SBEAMS is modular in design to allow the storage of various types of experiments in the system..."

Developer(s): Institute for Systems Biology, Seattle, Washington

Status: SBEAMS distribution v0.15 was available for downloading as of September 2003.

Availability:

- SBEAMS is licensed under General Public License (GPL).
- SBEAMS is available for downloading from http://www.sbeams.org/download/. A note on the Web page says that "Insufficient time has been devoted to making this product **easily** installable and usable at other locations."
- SBEAMS requires custom versions of the Data::ShowTable and GD::Graph Perl modules, which are available from the download page given above.
- The SBEAMS download includes schema diagrams (in the sbeams/doc directory) for the core, immunostain, microarray, phenoarray, proteomics, and SNP modules.
- The SBEAMS download includes SQL CREATE TABLE scripts (in the sbeams/lib/sql directory) for the core, Biosap, microarray, phenoarray, proteomics, and SNP modules.

Implementation:

- SBEAMS runs under the Linux and UNIX operating systems, but has not yet been tested using the Windows operating system.
- SBEAMS has been implemented in Perl and uses the Perl DBI module to connect to the database and Perl CGI scripts for the Web interface.
- SBEAMS uses the Microsoft SQL Server, but expects to support DB2 in the future.

Publication(s): No journal publications or conference papers to date.

Comments:

- SBEAMS has a core module that handles "user authentication, work group management, permissions management, simplified engine-indepenent SQL database access API, Web form abstraction, tabular data rendering, and much more".

- According to the SBEAMS Web page, microarray and proteomics modules have been implemented. The status of the other modules, for which there are schema diagrams or SQL scripts, is not clear.
- "The SBEAMS microarray module will be compliant with the emerging MAGE-OM/ML specification."

Appendix B: Data Standards for General Data Exchange

There are other data standards and markup languages that the data standards and markup languages in Appendix A are either based on, or include. These are listed below, together with other related, or relevant, markup languages.

AnIML - Analytical Information Markup Language

Link(s):

- http://animl.sourceforge.net/

CML - Chemical Markup Language

Link(s):

- http://xml.coverpages.org/cml.html
- http://www.xml-cml.org/

DICOM - Digital Imaging and Communications in Medicine

Link(s):

- A Nontechnical Introduction to DICOM from the Radiological Society of North America (RSNA)
- DICOM: An Introduction to the Standard from the Penn State Radiology Department
- DICOM Standard Status
- Links to information on DICOM and DICOM software tools from the ExPASy Web site

FMAS - Fluorescence Microscopy Annotation Schema

Link(s):

- http://murphylab.web.cmu.edu/services/FMAS/FMAS12.html

GAML - Generalized Analytical Markup Language

Link(s):

- Links to the GAML schema and an example data set

MathML - Markup Language for Mathematics

Link(s):

- http://www.w3.org/Math/
- http://www.w3.org/TR/REC-MathML

OME - Open Microscopy Environment

Link(s):

- http://tatooine.lbl.gov/www.openmicroscopy.org/
- http://tatooine.lbl.gov/cvs.openmicroscopy.org.uk/

SpectroML - Markup Language for Spectroscopic Data

Link(s):

- Description of SpectroML from XML Cover Pages
- Links to the DTD, XSD, and XSL for SpectroML as well as a code sample
- Description of SpectroML from NIST
- NIST FTP site for SpectroML

UnitsML - Units Markup Language

Link(s):

- http://unitsml.nist.gov/
- Draft schema for UnitsML

Appendix C: Glossary

Acronyms of data standards and markup languages defined in Appendices B and C are not included in this glossary.

AOHUPO: Asia Oceania Human Proteome Organisation (http://www.hupo.org/aohupo/)

ASTM: American Society for Testing and Materials (http://www.astm.org/)

BBSRC: Biotechnology and Biological Sciences Research Council (UK) (http://www.bbsrc.ac.uk/)

BIOSAP: Blast Integrated Oligonucleotide Selection Accelerator Package (http://biosap.sourceforge.net/)

CODATA: Committee on Data for Science and Technology (http://www.codata.org/)

COGEME: Consortium for the Functional Genomics of Microbial Eukaryotes (http://www.cogeme.man.ac.uk/)

DTD: Document Type Definition (http://www.w3.org/TR/REC-html40/sgml/dtd.html)
DTDs are used to define the structure and the legal elements of an XML document. A tutorial on how to write a DTD may be found at http://www.w3schools.com/dtd/.

ExPASy: Expert Protein Analysis System (http://us.expasy.org/)

FACS: Fluorescence Activated Cell Sorting

GPL: General Public License (http://www.gnu.org/licenses/gpl.html/)

HUPO: Human Proteome Organisation (http://www.hupo.org/)

I3C: Interoperable Informatics Infrastructure Consortium

IEF: isoelectric focusing

JDK:Java Development Kit (http://java.sun.com/)

ML: Markup Language

MALDI: matrix-assisted laser desorption/ionization

MGED: Microarray Gene Expression Data Society (http://www.mged.org/, http://www.mged.org/Mission/index.html/)

MySQL: open source database

netCDF: network Common Data Form (http://www.unidata.ucar.edu/packages/netcdf/)
"NetCDF (network Common Data Form) is an interface for array-oriented data access and a library that provides an implementation

of the interface. The netCDF library also defines a machine-independent format for representing scientific data. Together, the interface, library, and format support the creation, access, and sharing of scientific data. The netCDF software was developed at the Unidata Program Center in Boulder, Colorado."

NIST: National Institute of Standards and Technology (http://www.nist.gov/)

OMG: Object Management Group (http://www.omg.org/)

PAGE: polyacrylamide gel electrophoresis

RDBMS: relational database management system

SDS: sodium dodecyl sulphate

SMD: Stanford Microarray Database (http://genome-www5.stanford.edu/)

SNP: single nucleotide polymorphism

SQL: Structured Query Language

UML: Unified Modeling Language (http://www.omg.org/uml/)
UML is a visual language originally developed to describe software systems, but now widely used for data modeling, database design, and use case design. There are many resources available for learning about UML. The OMG UML page has links to UML resources, including an introduction to OMG UML. For examples of different kinds of UML diagrams, see the introduction by Ian Moraes or the Agile Modeling Web site. The Objects by Design Web site has a list of UML software products listed by platform and price.

W3C: World Wide Web Consortium (http://www.w3c.org/)

XML: eXtensible Markup Language. (http://www.w3c.org/XML/)
A quick overview of XML may be found at http://www.w3.org/XML/1999/XML-in-10-points.html. There are many online resources for learning about XML, some of which are listed on the W3C Web site. See http://www.w3schools.com/xml/default.asp for example.

Compilation of Data Formats and XMLs Applicable to Scientific and Analytical Data

Genomes To Life Data Standards Working Group
Argonne National Laboratory | BIATECH | Harvard University | Howard Hughes Medical Institute Lawrence Berkeley National Laboratory | Los Alamos National Laboratory | Oak Ridge National Laboratory Pacific Northwest National Laboratory University of California at Berkeley | University of California at San Diego

Contents

Contributing Members

John Ambrosiano, ambro@lanl.gov
Computer and Computational Sciences
Los Alamos National Laboratory

Adam Arkin, *Chairperson*, APArkin@lbl.gov
Physical Biosciences Division
Lawrence Berkeley National Laboratory
Department of Bioengineering
University of California, Berkeley
Howard Hughes Medical Institute

Gyorgy Babnigg, GBabnigg@anl.gov
Biosciences Division
Argonne National Laboratory

Ed Frank, EFrank@mcs.anl.gov
Mathematics and Computer Science Division
Argonne National Laboratory

Carol Giometti, CSGiometti@anl.gov
Biosciences Division
Argonne National Laboratory

Janet Jacobsen, JSJacobsen@lbl.gov
Institute for Quantitative Biomedical Research
University of California, Berkeley

Nagiza Samatova, SamatovaN@ornl.gov
Computer Science and Mathematics Division
Oak Ridge National Laboratory

Nancy Slater, NASlater@lbl.gov
Physical Biosciences Division
Lawrence Berkeley National Laboratory

Ron Taylor, Ronald.Taylor@pnl.gov
Biological Sciences Division
Pacific Northwest National Laboratory

Other members of the GTL Data Standards Working Group are

- Eric Alm, Physical Biosciences Division, Lawrence Berkeley National Laboratory
- Eugene Kolker, BIATECH
- Bertram Ludaescher, San Diego Supercomputer Center, University of California, San Diego
- George Michaels, Biological Sciences, Pacific Northwest National Laboratory
- D. William Nguyen, Harvard Medical School, Harvard University
- Bahram Parvin, Computational Research Division, Lawrence Berkeley National Laboratory
- Ed Uberbacher, Life Sciences Division, Oak Ridge National Laboratory

Objective

The purpose of this document is to provide background information on data formats and XMLs developed for scientific and analytical data. For the purpose of this document, analytical data refers to data collected from laboratory instruments. Analytical data is of course a subset of scientific data, which include data from many other sources, *e.g.,* meteorological, earth sciences, space sciences, *etc.* This document also discusses several ways in which binary data can be handled using XML.

Scope

A companion document to this one, Compilation of Data Standards for Proteomic and Transcriptomic Experimental Data [1], describes data standards specific to proteomic and transcriptomic experiments and implementations of those standards. The present document covers data formats and XMLs that are applicable to a broader range of scientific data sets and that may be applicable to certain types of data sets generated by GTL experiments. The list of data formats and XMLs included in this document is not exhaustive, but contains those that are most widely used or are the products of a standards committee.

Data Formats for Scientific Data

The problems associated with the transport and exchange of large data sets have been known to researchers in the space sciences, meteorology, and other physical sciences for many decades. These problems include transporting large amounts of data electronically, as well as standardizing data formats so that software written by one agency or organization will be applicable to data sets generated by other institutions. The situation with analytical data from laboratory experiments, as opposed to large field experiments (*e.g.*, air dispersion experiments) or monitoring large-scale phenomena (*e.g.*, atmospheric ozone, meteorological conditions), is somewhat different because in the former case, there are more experimental factors that need to be recorded as part of the data set. The following two sections briefly describe data formats for both types of data.

Formats for General Scientific Data

There are three well known data formats for scientific data: Common Data Format (CDF), Hierarchical Data Format (HDF), and Network Common Data Format (netCDF). They were developed with similar objectives in mind, namely the ability to store and manipulation large amounts of structured numeric data. All are self-describing in the sense that they include some description of the data they contain, that is, information about independent variables and the dimension of the data set(s) included. *Note that the 'metadata' included in the context of CDF, HDF, and netCDF describes the attributes of the data in the file, not* *experimental conditions such as type of equipment used, operating parameters, etc.* CDF, HDF, and netCDF were developed by different organizations for different applications, and each has a software library that provides access to files written in the data format. Some of the differences between CDF, HDF, and netCDF are described at the end of this section.

CDF [2] was developed by the National Space Science Data Center (NSSDC). CDF has two file format options. In the first option, all information about the data, that is, the metadata, and the data values are stored in one file. In the second option, the metadata is stored in one file, and there is a file for the data associated with each variable in the metadata file. One attractive feature of CDF for applications involving gridded data is that it provides a mechanism for storing information about gridded data without having to store all of the values of the grid. In addition, CDF allows for compression for a specified variable in a CDF file (but only when the single file format option is used); several compression algorithms are provided by the CDF library.

The core CDF library includes C, Fortran, and Java APIs. A Perl API is available separately. The APIs provide functions to create and query existing CDF files.

HDF [3] was developed by the National Center for Supercomputing Applications (NCSA). HDF5 was developed to "address the data management needs of scientists and engineers working in high performance, data intensive computing environments. As a result, the HDF5 library and format emphasize storage and I/O efficiency." NCSA currently supports two HDF formats, HDF4.x and HDF5, that are different and not compatible. A conversion utility exists to convert HDF4 files to HDF5 files. HDF5 can be partly converted to HDF4 files as well. Because NCSA recommends the use of HDF5, only HDF5 is described here.

HDF5 is a file format for storing scientific data that organizes the data into two kinds of objects: groups and data sets. Groups provide a structure for organizing different types of data sets (*e.g.* arrays, gridded data, images, *etc.*). A group may contain other groups or data sets. A group also includes a list of its attributes. A data set contains elements of data together with a description of the attributes of the data.

NCSA provides a Java Native Interface (JNI) to HDF5, as well as several command-line utilities that display the contents of an HDF5 file, compare two HDF5 files, import data into an HDF5 file, and generate XML output from an HDF5 file. NCSA has developed a DTD for HDF5 and plans to develop an XML Schema.

NetCDF [4] began as a reimplementation of the CDF library using XDR (eXternal Data Representation) [5] to provide a machine-independent data representation. NetCDF was developed by Unidata, a group of researchers in atmospheric, oceanic, and earth sciences. Unidata is hosted by the University Corporation for

Atmospheric Research (UCAR), which is funded by NSF, NOAA, NASA, DOD, DOE, and other government agencies. A netCDF file consists of two parts, a header that describes the dimensions and attributes of the variables whose data is contained in the second part. NetCDF is widely used because the data format is machine-independent, because of the relative ease of accessing netCDF files using utilities included in the netCDF distribution, and because so much other software has been developed to display and manipulate netCDF files [6]. NetCDF supports only structured grid data and is limited to two gigabytes per file.

Interfaces to netCDF have been written in C, C++, Fortran90, MATLAB, Java, Perl, Python, and Ruby.

Two notable differences between CDF and netCDF are that the CDF file format has both single file and multiple file options, whereas netCDF is restricted only to single files; and netCDF uses named dimensions (*e.g.*, TEMP[x, y, z, t]) whereas CDF indicates the dimensionality of a variable using 'true' or 'false', *i.e.*, TEMP[true, true, false, true]. As described above, HDF, in contrast to CDF, provides a hierarchical structure consisting of groups and data sets for storing data. More information about differences among the three formats may be found at [7, 8, 9]. Note that IBM's 3-D visualization package, Data Explorer [10], includes support for data files in all three formats.

Formats for Analytical Data

Developed by the Analytical Instrument Association (AIA), the Analytical Data Interchange (ANDI) is a data standard for analytical instrument data. Over a dozen manufacturers of mass spectrometry and chromatography equipment are able to export data files using the ANDI data standard [11, 12]. The ANDI data format contains a header to capture information about the equipment used and experiment parameters and either the raw or processed data. ANDI is based on netCDF, which was described in the previous section. A reference document for the ANDI standard [13] may be purchased from the American Society for Testing and Materials (ASTM) [14].

XMLs for Scientific Data

Development of XML (eXtensible Markup Language) [15] began in 1996 and has been a recommendation of the World Wide Web Consortium (W3C) since February 1998 [16]. XML has become a widely used methodology for describing structured data, and several of the data formats in the previous section use XML to describe the metadata associated with data sets. These and other XML implementations are briefly described below.

XMLs for General Scientific Data

Two of the XMLs described in this section, CDF Markup Language (CDFML) and NetCDF Markup Language (NcML), are based on the data formats, CDF and netCDF, respectively, described in the previous section. **CDFML** [17] is an XML that describes both the metadata and data of a CDF file. Java utilities are available to translate from CDF to CDFML and *vice versa*. **NcML** [18] is an XML representation for only the header information of a netCDF file. NcML consists of three parts: the NcML Core Schema, the NcML Coordinate System, and the NcML Dataset.

NCSA also has developed a Document Type Definition (DTD) for **HDF5** and plans to develop an XML Schema in the future [19].

NASA recently has developed an XML for general scientific data called the eXtensible Data Format (XDF) [20, 21]. The key features of **XDF** are that it supports hierarchical data structures, multi-dimensional arrays, tables, and variable resolution (*i.e.*, field width and precision). In addition, data values can be encoded within an XDF file, or references to external files can be included. There are APIs written for XDF in both Java (stable release as of June 2002) and Perl (stable release in May 2003).

Another relatively new markup language for scientific applications is the eXtensible Scientific Interchange Language (XSIL) [22, 23]. **XSIL** is being developed within the framework of the Caltech Center for Advanced Computing Research, Projects and Collaborations, primarily for astronomical applications. Like an XDF file, an XSIL file can handle both tables and arrays and can contain data or may reference external files. XDF uses 'stream' elements that encode data as text ("local stream"), binary ("external stream"), 'bigendian', 'littleendian', or Base64. (Base64 encoding and Endianism are described in Appendix A and Appendix B, respectively.) XSIL also has been implemented as a Java object model and comes with a Java object browser called Xlook [24].

XMLs for Analytical Data

GAML (Generalized Analytical Markup Language) [25, [26] was developed by a company called Thermo Galactic to store and archive data from a range of analytical instrumentation. GAML was submitted to the XML.org Registry [27] in November 2001. Examples of GAML files for different kinds of instruments, including FTIR spectroscopy, Raman spectroscopy, (1D) NMR, gas chromatography, liquid chromatography, and mass spectrometry are available [25]. GAML uses Base64 (see Appendix A) encoding for binary data in order to preserve numerical precision.

Thermo Galactic has developed software to convert data from 150 different formats (*i.e.*, data from different analytical instruments) to GAML. The software is proprietary, however, requiring the purchase of a license to use the software (one license per data format) as well as an annual maintenance fee to upgrade the software should the format change.

SpectroML (Markup Language for Spectrometry) [28] has been proposed as an ASTM standard for spectroscopy data. The National Institute of Standards and Technology (NIST) [29] submitted SpectroML to the XML.org Registry in January 2002 [30]. The entry in the XML.org Registry entry contains links to the Document Type Definition (DTD), XML Schema Definition (XSD), and Extensible Stylesheet Language (XSL) for SpectroML.

The ASTM Subcommittee on Analytical Data Management (E13.15), which defines standards for "representing, managing, and interchanging analytical chemistry data including the implementation of technique specific information and application to instrument data interfaces", met in March 2003 to review GAML and SpectroML [31]. At that meeting, it was proposed that a new data exchange standard, Analytical Information Markup Language **AnIML** [32, 33], be developed based on the best features of GAML and SpectroML.

Handling Binary Data in XML

As noted earlier, interest in implementing data standards for experimental data in XML is based on the ability of XML to represent structured data. In the case of experimental data, this means being able to describe how an experiment was conducted, who conducted it, when it was conducted, why it was conducted, experimental factors, and equipment parameters. In other words, XML is very good at capturing an experiment's metadata. In contrast, the data formats discussed earlier (*e.g.,* netCDF) are better than XML for exchanging or storing large data sets, especially binary data sets, because they were developed specifically for such applications.

Nevertheless, it is possible to either include encoded binary data within an XML file or to place a link in an XML file to an external file that contains the data set. The format or method of accessing the external data set also would be included in the link element.

The following is an example of how to reference numeric data in an external file. In this case, there are two sets of data for the given experiment; both have been stored using the netCDF format. Being able to access the data would require netCDF software to read and interpret the data in the files.

```
<experiment>
  ...
  <data>
    <xlink href="ftp://gtl.lbl.gov/dataset-1.nc">
    <format>netcdf</format>
  </data>
```

```
    <data>
      <xlink href="ftp://gtl.lbl.gov/dataset-2.nc">
      <format>netcdf</format>
    </data>
  </experiment>
```

In the case of including binary data within an XML file, GAML, which was mentioned earlier, uses Base64 encoding (see Appendix A) to store binary data as ASCII strings within GAML files in order to preserve numerical precision [26]. The mzXML standard for mass spectrometry data uses Base64 encoding for peak data [34].

The ASTM E13.15 Working Group voted to adopt Base64 encoding for floating point numbers at its April 2003 meeting [35]. A recent working document by Peter J. Linstrom of the National Institute of Standards and Technology [36] describes a data model conforming to the ASTM E13.15 standard, for analytical chemistry data (specifically FITR and GC-MS data) that uses Base64 encoding with little Endian byte-ordering (see Appendix B) for 32-bit or 64-bit floating point numbers. Linstrom includes a Java program for encoding floating point numbers in Base64 and decoding encoded floating point numbers in the document.

The Binary XML Description Language, BinX was developed to describe the content and structure of binary files [37]. BinX is accompanied by a BinX software library written in C++ for reading and writing BinX files and the associated files of binary data. The BinX library is available as a pre-compiled library for Linux.

Discussion and Summary

This document reviews a number of data formats and XMLs for general scientific data as well as for analytical data. Many of the data formats and XMLs seem similar to one another making it difficult to determine which ones may be more likely to be applicable to data from GTL experiments. In addition, some of the data formats/XMLS are under development or are newly minted, and therefore, will be subject to revision in the coming months.

One factor to consider in choosing a data format or XML is what kind of software may be available to support that data format or XML, and how stable the software release is. NetCDF, for example, has been in use in the atmospheric sciences for many years and is supported by software utilities to read and write netCDF files. In contrast, mzXML was developed within the last year, and the mzXML software to convert mass spec data from vendor format to mzXML is a 'development' version.

It also is possible that no one single data format or XML will cover data from all GTL experiments, requiring instead the use of more than one. What is needed at this time is a comparison of the data standards/XMLs in this document by attempting to apply them to test cases, *i.e.*, data sets from GTL experiments.

Only by examining the details of the various data formats/XMLs through application to test cases will it become clear which data formats/XMLS are applicable to data from GTL experiments.

Acknowledgement

We are grateful to Frank Olken of the Scientific Data Management Research Group, Computational Research Division, at Lawrence Berkeley National Laboratory for taking the time to review this Web document and for his helpful suggestions on how to restructure it.

Date of Last Revision: January 5, 2004

Appendix A: Base64 Encoding

Base64 encoding [38] is an encoding scheme for the portable transport of binary data. In Base64 encoding, groups of 24 bits are represented as strings of four characters. What makes data encoded by the the Base64 scheme portable is that the Base64 alphabet consists of 65 characters that are represented identically in all versions of ISO 646, which includes US-ASCII, and all versions of EBCDIC.

The input group of 24 bits is formed by concatenating three groups each containing eight bits (*e.g.*, three eight-bit bytes). The 24 bits are divided into four groups of six bits each. Each group of six bits is translated into a single character according to the Base64 alphabet. A group of six bits can represent a number between 0 and 63, inclusive. In the Base64 alphabet, the numbers 0 through 25 map to the letters A through Z, the numbers 26 to 51 map to the letters a through z, the numbers 52 through 61 map to the digits 0 through 8, the number 62 maps to the character +, and 63 maps to the character / (see Table 1 in [37]). The 65th character in the Base64 alphabet is the equal sign (=), which is used to pad encoded strings when necessary.

As an example, if the 24-bit binary number, 00010010 11010110 10000111 is divided into four groups of six bits, the result is
000100 101101 011010 000111, which in decimal is 4 54 26 7. Translated into Base64, the result is E2aH (4->E, 54->2, 26->a, and 7->H). Translation of binary data into the Base64 alphabet is straightforward and converters exist in various programming languages (*e.g.*, Java [36], Perl [39], and Python [40]).

Appendix B: Endianism

Endianism refers to the order of the most significant byte in multi-byte numbers. In a big Endian system, the most significant byte is stored in the lowest address, and in a little Endian system, the most significant byte is stored in the highest address. For example, in binary notation, the number 1,234 would be represented as 10011010010. In a computer using 8-bit bytes and big Endian byte ordering, 1,234 would be 00000100 11010010. In a computer using little Endian byte ordering, 1,234 would be 11010010 00000100. It is essential to know the byte order in order to correctly interpret the values of numeric data represented in binary.

Intel processors, and thus the PCs that use Intel processors, use the little Endian byte order. Other computer systems that use the little Endian byte order are Cray and Digital Equipment Corporation (DEC) systems. Motorola processors use the big Endian byte order, as do Sun Sparc and Silicon Graphics, Inc. (SGI) Irix workstations. Users who exchange binary data written on different computer systems need to be aware that the byte order may differ between their systems.

> *The names, Little Endian and Big Endian, refer to political opponents in Jonathan Swift's Gulliver's Travels. The king of Lilliput decreed that his subjects, the Little Endians, break their eggs at the small end of the egg. The Big Endians rebelled against the king and broke their eggs at the big end of the egg.*

References

[1] GTL Data Standards Working Group. *Compilation of Data Standards for Proteomic and Transcriptomics Experimental Data*, January 2004. Available via the World Wide Web at http://vimss.lbl.gov/~jsjacobsen/GTL/datastds.html.

[2] National Space Science Data Center (NSSDC). *The Common Data Format (CDF)*. Information about CDF available via the World Wide Web at http://nssdc.gsfc.nasa.gov/cdf/cdf_home.html.

[3] National Center for Supercomputing Applications (NCSA). *Information, Support, and Software from the Hierarchical Data Format (HDF) Group of NCSA*. Available via the World Wide Web at http://hdf.ncsa.uiuc.edu.

[4] Unidata, University Corporation for Atmospheric Research. *NetCDF*. Information about netCDF is available via the World Wide Web at http://www.unidata.ucar.edu/packages/netcdf

[5] Sun Microsystems, Inc. Network Working Group. *XDR: External Data Representation Standard*, June 1987. Available via the World Wide Web at http://rfc1014.x42.com.

[6] Unidata, University Corporation for Atmospheric Research. *Software for Manipulating or Displaying NetCDF Data*. Available via the World Wide Web at http://www.unidata.ucar.edu/packages/netcdf/software.html.

[7] Shea, D.J., Worley, S.J., Stern, I.R., and Hoar, T.J. *An Introduction to Atmospheric and Oceanographic Datasets*, NCAR TECHNICAL NOTE NCAR/TN-404+IA, August 1994. See *Table B.1. Comparison of three scientific-data-management systems, Appendix B. Features of Common Scientific Data Formats; Access Information*. Available via the World Wide Web at http://www.cgd.ucar.edu/cas/tn404/text/tn404_app-b.html.

[8] National Space Science Data Center (NSSDC). *CDF Frequently Asked Questions*. See *8. What are the differences between CDF and NetCDF, and CDF and HDF?* available via the World Wide Web at http://nssdc.gsfc.nasa.gov/cdf/html/FAQ.html.

[9] Unidata, University Corporation for Atmospheric Research. *Frequently Asked Questions About netCDF*. See *What is the connection between netCDF and CDF?* and *What is the connection between netCDF and HDF?* available via the World Wide Web at http://www.unidata.ucar.edu/packages/netcdf/faq.html.

[10] IBM Research. *Open Visualization Data Explorer*. Information available via the World Wide Web at http://www.research.ibm.com/dx.

[11] High Performance Liquid Chromatography Users Group. *FAQ on AIA* includes information about ANDI. See http://hplcusersgroup.tripod.com/faq/AIA.html.

[12] Analytical Instrument Association (AIA). *Analyltical Data Interchange (ANDI)*. See the ANDI SourceForge site at http://sourceforge.net/projects/andi.

[13] American Society for Testing and Materials (ASTM) International. *E1947-98 Standard Specification for Analytical Data Interchange Protocol for Chromatographic Data*. Available for purchase at http://www.astm.org/DATABASE.CART/PAGES/E1947.htm.

[14] American Society for Testing and Materials (ASTM) International. The URL for the ASTM Web site is http://www.astm.org.

[15] World Wide Web Consortium (W3C). *Extensible Markup Language (XML)*. Available via the World Wide Web at http://www.w3.org/XML.

[16] Bos, B. *XML in 10 points*, March 1999. Available via the World Wide Web at http://www.w3.org/XML/1999/XML-in-10-points.

[17] National Space Science Data Center (NSSDC). *CDF Embraces XML*. Information available via the World Wide Web at http://nssdc.gsfc.nasa.gov/cdf/html/cdf_xml.html.

[18] Unidata, University Corporation for Atmospheric Research. *The NetCDF Markup Language (NcML)*. Information available via the World Wide Web at http://www.unidata.ucar.edu/packages/netcdf/ncml.

[19] National Center for Supercomputing Applications (NCSA). *The HDF5 XML Information Page*. Available via the World Wide Web at http://hdf.ncsa.uiuc.edu/HDF5/XML.

[20] Organization for the Advancement of Structured Information Standards (OASIS). *Extensible Data Format (XDF)*. A description of XDF is available at the OASIS Cover Pages Web site at http://xml.coverpages.org/xdf.html.

[21] XML Group at the NASA Goddard Space Flight Center (GSFC). *eXtensible Data Format (XDF) Homepage*. Information available via the World Wide Web at http://xml.gsfc.nasa.gov/XDF/XDF_home.html

[22] Organization for the Advancement of Structured Information Standards (OASIS). *Extensible Scientific Interchange Language (XSIL)*. A description of XSIL is available at the OASIS Cover Pages Web site at http://xml.coverpages.org/xsil.html.

[23] Center for Advanced Computing Research (CACR) at the California Institute of Technology. *XSIL: Extensible Scientific Interchange Language*. Information available via the World Wide Web at http://www.cacr.caltech.edu/SDA/xsil.

[24] Williams, R. *XSIL: Java/XML for Scientific Data*, June 2000. The paper is available via the World Wide Web at http://www.cacr.caltech.edu/projects/xsil/xsil_spec.pdf.

[25] Thermo Galactic. *Generalized Analytical Markup Language (GAML)*. Information about GAML is available via the World Wide Web at http://www.gaml.org.

[26] Duckworth, J. *An XML-Based File Format for Archival Storage of Analytical Instrument Data*, October 2001. http://www.gaml.org/Documentation/XML Analytical Archive Format.pdf

[27] Organization for the Advancement of Structured Information Standards (OASIS). Information about the *XML.org Registry* is available at

http://www.xml.org/xml/registry.jsp. The XML.org Registry is "a central clearinghouse for developers and standards bodies to publicly submit, publish and exchange XML schemas, vocabularies and related documents".

[28] Kramer, G. *Standards for Exchange of Instrument Data and NIST Chemical Reference Data - SpectroML*, November 2002. Available on the World Wide Web at http://www.mel.nist.gov/div826/msid/sima/03_spectro.html.

[29] National Institute of Standards and Technology (NIST). The URL for the NIST Web site is http://www.nist.org.

[30] National Institute of Standards and Technology (NIST). The DTD, XSL, XSD, and a sample file for SpectroML are available on the World Wide Web at http://hercules.xml.org/xml/schema/e0543eb1.

[31] AnIML Working Group, American Society for Testing and Materials (ASTM) E13.15 Committee. *Minutes of the ASTM E13.15 Committee Meeting*, March 11, 2003, Orlando, Florida. The minutes are available via the World Wide Web at http://animl.sourceforge.net/E13.15%20Minutes%203_11_03.pdf

[32] AnIML Working Group, American Society for Testing and Materials (ASTM) E13.15 Committee. *Analytical Information Markup Language "AnIML"*. Information available via the World Wide Web at http://animl.sourceforge.net.

[33] National Institute of Standards and Technology (NIST). *NIST AnIML Proposal*. Available via the World Wide Web at http://animl.sourceforge.net/nist.

[34] Pedrioli, P., Eng, J., Hubley, R., Pratt, B., Nillson, E., Taylor, A. and Aebersold, R. *Creation of an open standard file format for the representation of MS data*, presented at the 51st ASMS Conference, June 8-12, 2003, Montreal, Canada. The abstract is available via the World Wide Web at http://sashimi.sourceforge.net/extra/abstract.pdf. mzXML is described in more detail in [1].

[35] AnIML Working Group of the American Society for Testing and Materials (ASTM) E13.15 Committee. *Minutes of the ASTM E13.15 Committee Meeting*, April 23-24, 2003, Philadelphia, Pennsylvania. The minutes are available via the World Wide Web at http://animl.sourceforge.net/E13.15%20Minutes%204_23_03.pdf

[36] Linstrom, P. J. *Examples of a data model for ASTM E13.15*, June 20, 2003. Available via the World Wide Web at http://animl.sourceforge.net/examples/examples.pdf.

[37] Escience Data Information and Knowledge Tranformation. Information on BinX and the BinX library is available via the World Wide Web at http://www.edikt.org/binx.

[38] Borenstein, N. and Freed, N. *MIME (Multipurpose Internet Mail Extensions) Part One*, September 1993. See *Section 5.2. Base64 Content-Transfer-Encoding*. Available via the World Wide Web at http://www.mhonarc.org/~ehood/MIME/1521/05_Content-Transfer-Encoding.html.

[39] Perl 5.6 Documentation. *MIME::Base64*. http://www.perldoc.com/perl5.6/lib/MIME/Base64.html.

[40] Python Library Reference. *12.12 binascii -- Convert between binary and ASCII*. Information available via the World Wide Web at http://www.python.org/doc/1.6/lib/module-binascii.html.

Berkeley BioSpice / Arkin Lab
Lawrence Berkeley National Laboratory
University of California, Berkeley

Title

BioDB Schemas for Experiment and Protocol Metadata

Introduction

This document contains several figures that describe database schemas for storing information about experiments and experiment protocols. The objective of the schema designs is to capture information that is general enough to apply to most experiments and protocols, but specific enough to be of value to users searching the database for experiments and data related to organisms or biological processes of interest. In addition, these schemas should be compatible with schemas and standards developed for specific kinds of experimental methods and data such as MIAME (Mimimum Information About Microarray Experiments) and PEDRo (Proteomics Experimental Data Repository).

In the following section, the first figure provides an overview of the experiment and protocol schemas. Figures that follow show the fields of the tables that comprise the experiment or the protocol schemas. Because these schemas are part of BioDB, they use existing tables in BioDB for storing names, dates, addresses, organism references, *etc.* Fields in tables that end in "_objid" and are of type UNIQUEIDENTIFIER refer to other tables in BioDB. BioDB is based on the BIND/NCBI schema.

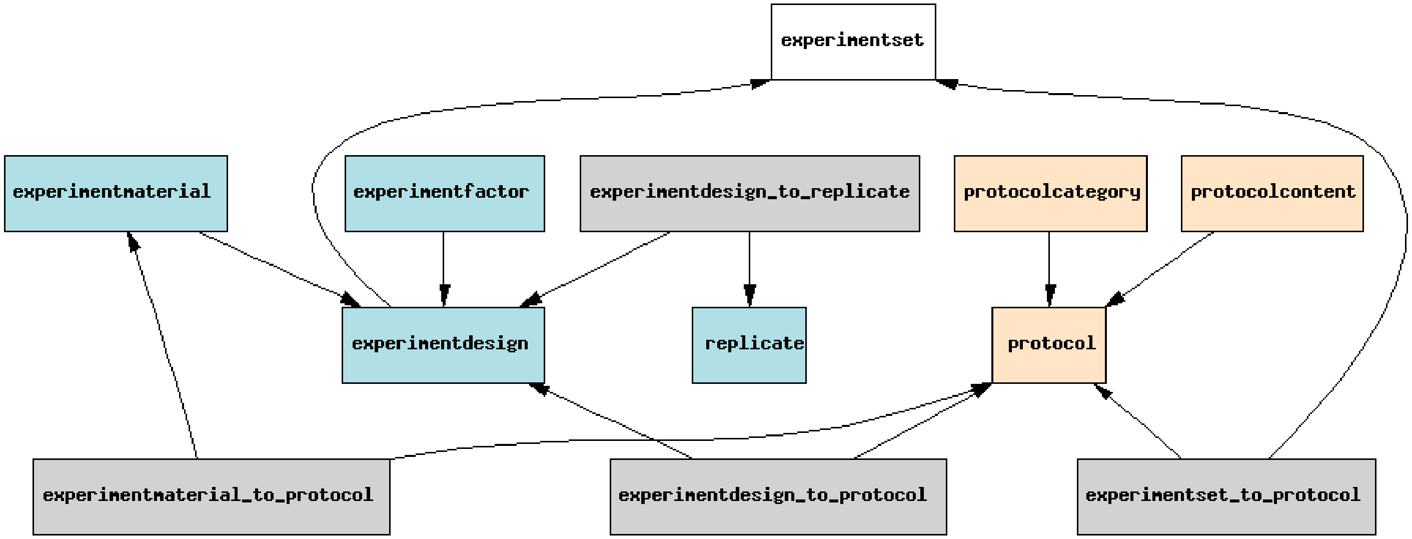
Overview of Experiment and Protocol Schemas

The **experimentset** table is the top-level table for experiment metadata and contains information that pertains to *all* experiments in a given *set* of experiments. An individual experiment is defined as a single treatment for a given set of experimental conditions. The **experimentdesign** and associated tables contain the metadata for an individual experiment as defined above.

Experiment protocols may be associated with a set of experiments, with an individual experiment, or with preparation of a sample. An individual experiment or set of experiments may have one or more protocols associated with it depending on how the protocols are written.

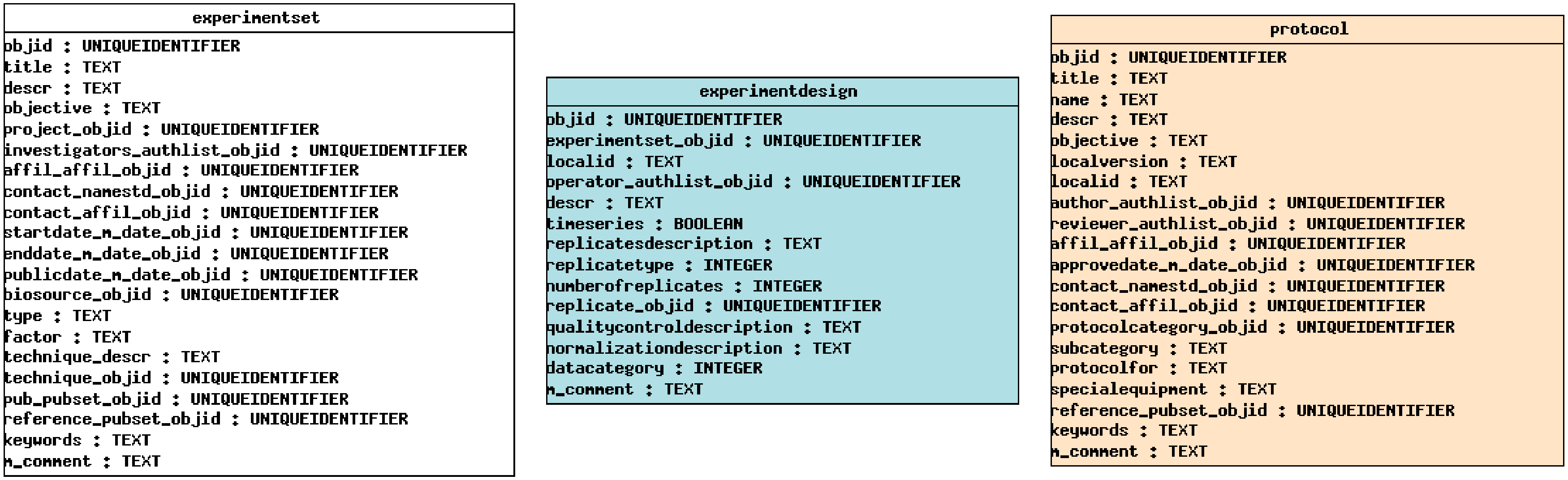
Association or linking tables are shown in light gray. Association tables are used to provide a one-to-many or many-to-many linking of tables. For example, the **experimentdesign_to_protocol** table provides a way to link an experiment with several protocols that may be involved in performing the experiment.

Figure 1. Overview of Experiment and Protocol Schemas


experimentset
experimentmaterial
experimentfactor
experimentdesign_to_replicate
protocolcategory
protocolcontent
experimentdesign
replicate
protocol
experimentmaterial_to_protocol
experimentdesign_to_protocol
experimentset_to_protocol

The figure below shows the fields of the primary tables in the experiment and protocol schemas.

Figure 2. Primary Tables of the Experiment and Protocol Schemas



experimentset
objid : UNIQUEIDENTIFIER
title : TEXT
descr : TEXT
objective : TEXT
project_objid : UNIQUEIDENTIFIER
investigators_authlist_objid : UNIQUEIDENTIFIER
affil_affil_objid : UNIQUEIDENTIFIER
contact_namestd_objid : UNIQUEIDENTIFIER
contact_affil_objid : UNIQUEIDENTIFIER
startdate_m_date_objid : UNIQUEIDENTIFIER
enddate_m_date_objid : UNIQUEIDENTIFIER
publicdate_m_date_objid : UNIQUEIDENTIFIER
biosource_objid : UNIQUEIDENTIFIER
type : TEXT
factor : TEXT
technique_descr : TEXT
technique_objid : UNIQUEIDENTIFIER
pub_pubset_objid : UNIQUEIDENTIFIER
reference_pubset_objid : UNIQUEIDENTIFIER
keywords : TEXT
m_comment : TEXT

experimentdesign
objid : UNIQUEIDENTIFIER
experimentset_objid : UNIQUEIDENTIFIER
localid : TEXT
operator_authlist_objid : UNIQUEIDENTIFIER
descr : TEXT
timeseries : BOOLEAN
replicatesdescription : TEXT
replicatetype : INTEGER
numberofreplicates : INTEGER
replicate_objid : UNIQUEIDENTIFIER
qualitycontroldescription : TEXT
normalizationdescription : TEXT
datacategory : INTEGER
m_comment : TEXT

protocol
objid : UNIQUEIDENTIFIER
title : TEXT
name : TEXT
descr : TEXT
objective : TEXT
localversion : TEXT
localid : TEXT
author_authlist_objid : UNIQUEIDENTIFIER
reviewer_authlist_objid : UNIQUEIDENTIFIER
affil_affil_objid : UNIQUEIDENTIFIER
approvedate_m_date_objid : UNIQUEIDENTIFIER
contact_namestd_objid : UNIQUEIDENTIFIER
contact_affil_objid : UNIQUEIDENTIFIER
protocolcategory_objid : UNIQUEIDENTIFIER
subcategory : TEXT
protocolfor : TEXT
specialequipment : TEXT
reference_pubset_objid : UNIQUEIDENTIFIER
keywords : TEXT
m_comment : TEXT

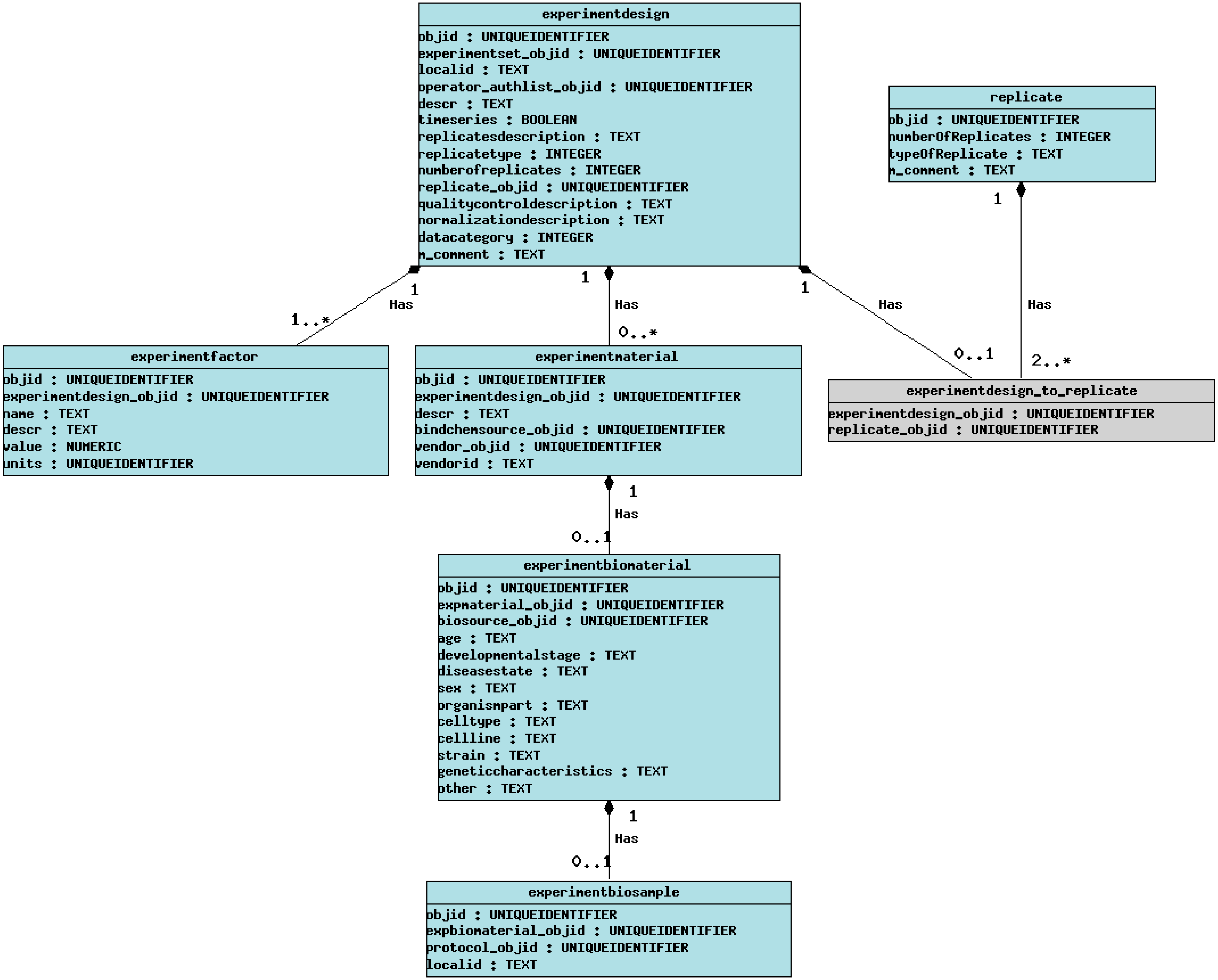
Experiment Design Schema

The **experimentdesign** table contains information about a single experiment in a set of experiments. The **experimentfactor** and **experimentmaterial** tables provide more detailed or quantitative information about the design of the experiment. Each entry in the **experimentdesign** table may correspond to more than one entry in the **experimentfactor** and **experimentmaterial** tables.

The **experimentmaterial** table applies to both biological and non-biological materials used in an experiment. If the material is biological in nature, then the **experimentbiomaterial** table provides more attributes to characterize it. If the biological material is subjected to a protocol in order to create a biological sample for the experiment, then the **experimentbiosample** table provides fields for both the protocol identifier and the identifier used locally by the lab producing the sample.

The **replicate** table provides a way to link together experiments that are replicates of one another.

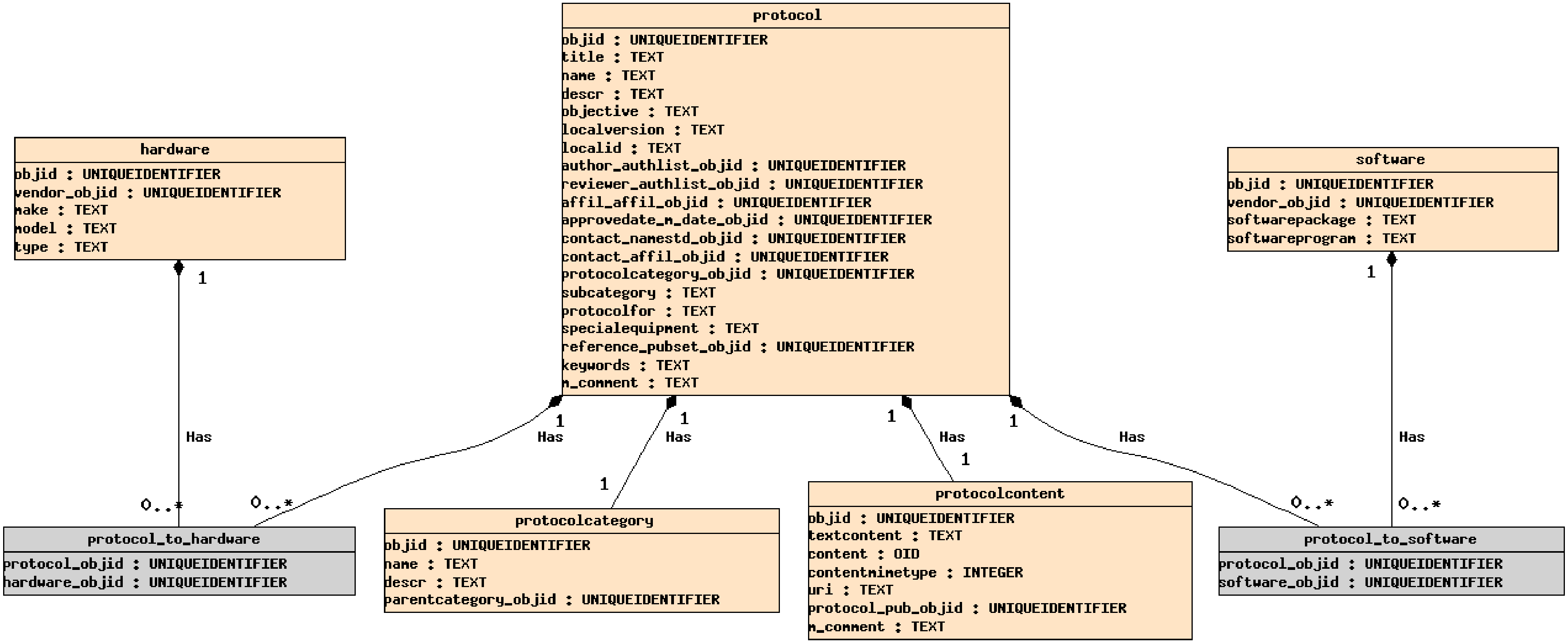
Figure 3. Experiment Design Schema


experimentdesign
objid : UNIQUEIDENTIFIER
experimentset_objid : UNIQUEIDENTIFIER
localid : TEXT
operator_authlist_objid : UNIQUEIDENTIFIER
descr : TEXT
timeseries : BOOLEAN
replicatesdescription : TEXT
replicatetype : INTEGER
numberofreplicates : INTEGER
replicate_objid : UNIQUEIDENTIFIER
qualitycontroldescription : TEXT
normalizationdescription : TEXT
datacategory : INTEGER
m_comment : TEXT
replicate
objid : UNIQUEIDENTIFIER
numberOfReplicates : INTEGER
typeOfReplicate : TEXT
m_comment : TEXT
1
1
Has
1..*
1
Has
0..*
1
Has
0..1
Has
2..*
experimentfactor
objid : UNIQUEIDENTIFIER
experimentdesign_objid : UNIQUEIDENTIFIER
name : TEXT
descr : TEXT
value : NUMERIC
units : UNIQUEIDENTIFIER
experimentmaterial
objid : UNIQUEIDENTIFIER
experimentdesign_objid : UNIQUEIDENTIFIER
descr : TEXT
bindchemsource_objid : UNIQUEIDENTIFIER
vendor_objid : UNIQUEIDENTIFIER
vendorid : TEXT
experimentdesign_to_replicate
experimentdesign_objid : UNIQUEIDENTIFIER
replicate_objid : UNIQUEIDENTIFIER
1
Has
0..1
experimentbiomaterial
objid : UNIQUEIDENTIFIER
expmaterial_objid : UNIQUEIDENTIFIER
biosource_objid : UNIQUEIDENTIFIER
age : TEXT
developmentalstage : TEXT
diseasestate : TEXT
sex : TEXT
organismpart : TEXT
celltype : TEXT
cellline : TEXT
strain : TEXT
geneticcharacteristics : TEXT
other : TEXT
1
Has
0..1
experimentbiosample
objid : UNIQUEIDENTIFIER
expbiomaterial_objid : UNIQUEIDENTIFIER
protocol_objid : UNIQUEIDENTIFIER
localid : TEXT

Protocol Schema

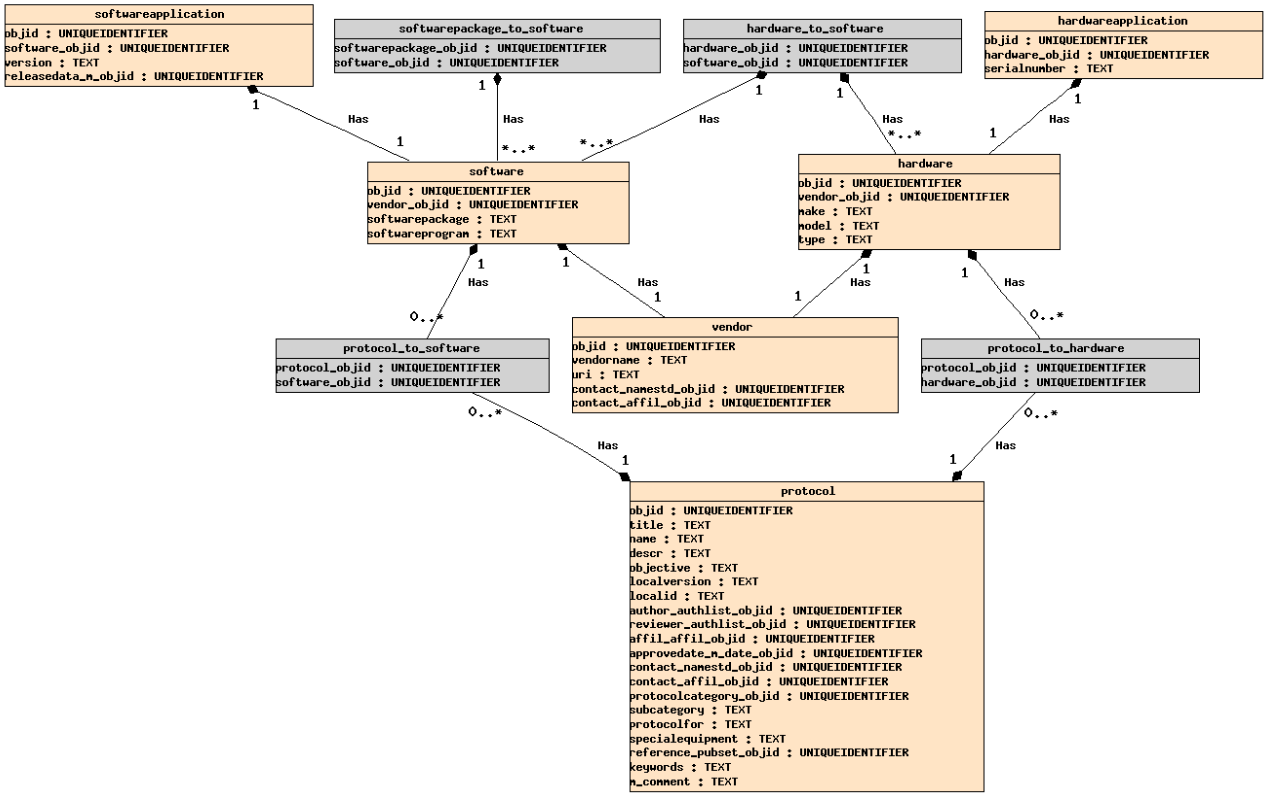
The first figure belows shows the main tables of the protocol schema. The **protocolcontent** table allows a user to store the protocol in the database in different formats, *e.g.,* as text or as a pdf file (pdf files are stored as binary large objects), or to store references to protocols, either as a reference to a publication or to a URI. Though it is preferable to store the protocol in the database, some protocols are taken from manufacturer's manuals.

Figure 4. Protocol Schema


protocol
objid : UNIQUEIDENTIFIER
title : TEXT
name : TEXT
descr : TEXT
objective : TEXT
localversion : TEXT
localid : TEXT
author_authlist_objid : UNIQUEIDENTIFIER
reviewer_authlist_objid : UNIQUEIDENTIFIER
affil_affil_objid : UNIQUEIDENTIFIER
approvedate_m_date_objid : UNIQUEIDENTIFIER
contact_namestd_objid : UNIQUEIDENTIFIER
contact_affil_objid : UNIQUEIDENTIFIER
protocolcategory_objid : UNIQUEIDENTIFIER
subcategory : TEXT
protocolfor : TEXT
specialequipment : TEXT
reference_pubset_objid : UNIQUEIDENTIFIER
keywords : TEXT
m_comment : TEXT
hardware
objid : UNIQUEIDENTIFIER
vendor_objid : UNIQUEIDENTIFIER
make : TEXT
model : TEXT
type : TEXT
software
objid : UNIQUEIDENTIFIER
vendor_objid : UNIQUEIDENTIFIER
softwarepackage : TEXT
softwareprogram : TEXT
1
Has
0..*
1
Has
0..*
1
Has
1
1
Has
1
1
Has
0..*
1
Has
0..*
protocol_to_hardware
protocol_objid : UNIQUEIDENTIFIER
hardware_objid : UNIQUEIDENTIFIER
protocolcategory
objid : UNIQUEIDENTIFIER
name : TEXT
descr : TEXT
parentcategory_objid : UNIQUEIDENTIFIER
protocolcontent
objid : UNIQUEIDENTIFIER
textcontent : TEXT
content : OID
contentmimetype : INTEGER
uri : TEXT
protocol_pub_objid : UNIQUEIDENTIFIER
m_comment : TEXT
protocol_to_software
protocol_objid : UNIQUEIDENTIFIER
software_objid : UNIQUEIDENTIFIER

The **hardware** and **software** tables and related tables were taken almost directly from ArrayExpress. The **hardwareapplication** and **softwareapplication** tables identify particular pieces of equipment by serial number and software packages by version and release date, respectively. The **softwarepackage_to_software** table provides a way to describe the software requirements of a software package.

Figure 5. Hardware and Software Tables in Protocol Schema


softwareapplication
objid : UNIQUEIDENTIFIER
software_objid : UNIQUEIDENTIFIER
version : TEXT
releasedata_m_objid : UNIQUEIDENTIFIER
softwarepackage_to_software
softwarepackage_objid : UNIQUEIDENTIFIER
software_objid : UNIQUEIDENTIFIER
hardware_to_software
hardware_objid : UNIQUEIDENTIFIER
software_objid : UNIQUEIDENTIFIER
hardwareapplication
objid : UNIQUEIDENTIFIER
hardware_objid : UNIQUEIDENTIFIER
serialnumber : TEXT
software
objid : UNIQUEIDENTIFIER
vendor_objid : UNIQUEIDENTIFIER
softwarepackage : TEXT
softwareprogram : TEXT
hardware
objid : UNIQUEIDENTIFIER
vendor_objid : UNIQUEIDENTIFIER
make : TEXT
model : TEXT
type : TEXT
protocol_to_software
protocol_objid : UNIQUEIDENTIFIER
software_objid : UNIQUEIDENTIFIER
vendor
objid : UNIQUEIDENTIFIER
vendorname : TEXT
uri : TEXT
contact_namestd_objid : UNIQUEIDENTIFIER
contact_affil_objid : UNIQUEIDENTIFIER
protocol_to_hardware
protocol_objid : UNIQUEIDENTIFIER
hardware_objid : UNIQUEIDENTIFIER
protocol
objid : UNIQUEIDENTIFIER
title : TEXT
name : TEXT
descr : TEXT
objective : TEXT
localversion : TEXT
localid : TEXT
author_authlist_objid : UNIQUEIDENTIFIER
reviewer_authlist_objid : UNIQUEIDENTIFIER
affil_affil_objid : UNIQUEIDENTIFIER
approvedate_m_date_objid : UNIQUEIDENTIFIER
contact_namestd_objid : UNIQUEIDENTIFIER
contact_affil_objid : UNIQUEIDENTIFIER
protocolcategory_objid : UNIQUEIDENTIFIER
subcategory : TEXT
protocolfor : TEXT
specialequipment : TEXT
reference_pubset_objid : UNIQUEIDENTIFIER
keywords : TEXT
m_comment : TEXT
Has
1
..
0..*

List of Figures

See Also

See the following links for more detailed information about ArrayExpress, BIND, MIAME, and PEDRo.

ArrayExpress	**http://www.ebi.ac.uk/arrayexpress**
Biomolecular Interaction Network Database (BIND)	**http://www.bind.ca**
Minimum Information about a Microarray Experiment (MIAME)	**http://www.mged.org/Workgroups/MIAME/miame.html**
Proteomics Experimental Data Repository (PEDRo)	**http://pedro.man.ac.uk/home.shtml**

Author: Janet Jacobsen <jsjacobsen@lbl.gov>

Date of Last Revision: January 31, 2004